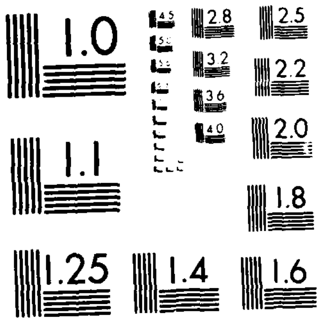
AD-A194 336 ADVANCED COMMUNICATION SYSTEM ENGINEERING PROCEEDINGS WORKSHOP HELD IN SE..(U) UNIVERSITY OF SOUTHERN CALIFORNIA LOS ANGELES COMMUNICATION S.. R A SCHOLTZ 1/5

UNCLASSIFIED MAR 88 CSI-88-03-02 ARO-24913.1-EL-CF F/G 25/5 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

INSTITUTE

ADVANCED COMMUNICATION SYSTEM ENGINEERING

May 26-29, 1987

CSI-88-03-02

2

ADVANCED COMMUNICATION SYSTEM ENGINEERING

May 26-29, 1987

CSI-88-03-02

UNCLASSIFIED MASTER COPY - FOR REPRODUCTION PURPOSES

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY 2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) ARO 24913.1-EL-CF
6a. NAME OF PERFORMING ORGANIZATION U of Southern California 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION U. S. Army Research Office
6c. ADDRESS (City, State, and ZIP Code) Los Angeles, CA 90089-0272	7b. ADDRESS (City, State, and ZIP Code) P. O. Box 12211 Research Triangle Park, NC 27709-2211
8a. NAME OF FUNDING / SPONSORING ORGANIZATION U. S. Army Research Office 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DAAL03-87-G-0101
8c. ADDRESS (City, State, and ZIP Code) P. O. Box 12211 Research Triangle Park, NC 27709-2211	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification)
Advanced Communication System Engineering

12. PERSONAL AUTHOR(S)
Robert A. Scholtz

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM 3/16/87 TO 3/15/88	March 1988	451

16. SUPPLEMENTARY NOTATION
The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Advanced Communication Systems, Communications, Communication Systems Design, Optical Processing, Computer-Aided Modeling

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The workshop had as its objective "to determine the state-of-the-art and explore future trends in design tools for communication system engineers, and to discuss the potential for utilization of optical processing in communication systems." These topics are all related to solving the large-scale communication system design problems of the future.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code) / 22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Five panels were formed, each manned by experts from industry and universities and in some cases government representatives as well. The panels were: (a) *Design tools of the future*, aimed at trying to predict the needs of future communication system designers, (b) *Computer-aided modelling, analysis, and design of communication systems*, to discuss tools for the system engineer, (c) *The status of optical signal processing*, to propose new applications of optical processing, (d) *A review of the state-of-the-art in communication system design*, to describe the critical problems presently confronting designers, and (e) *A group discussion: the role of government, industry, and universities in advanced communication system engineering*, to bring out the cooperative and managerial problems that must be solved to make real improvements in the communication system design process. Each panel consisted of formal presentations and discussions.

ADVANCED COMMUNICATION SYSTEM ENGINEERING

The proceedings of a workshop sponsored by the

U.S. ARMY RESEARCH OFFICE
P.O. Box 12211
Research Triangle Park
North Carolina 27709

under Contract No. DAAL03-87-G-0101

and the

COMMUNICATION SCIENCES INSTITUTE
Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089-0272

May 26-29, 1987

TABLE OF CONTENTS

PREFACE

This document contains the proceedings of the workshop "Advanced Communication System Engineering," held May 26-29, 1987, in Sedona, Arizona. Sponsored by the Army Research Office (under Contract DAAL03-87-G-0101) and organized by the Communication Sciences Institute of the University of Southern California, the workshop had as its objective "to determine the state-of-the-art and explore future trends in design tools for communication system engineers, and to discuss the potential for utilization of optical processing in communication systems." These topics are all related to solving the large-scale communication system design problems of the future.

Five panels were formed, each manned by experts from industry and universities and in some cases government representatives as well. The panels were: (a) *Design tools of the future*, aimed at trying to predict the needs of future communication system designers, (b) *Computer-aided modelling, analysis, and design of communication systems*, to discuss tools for the system engineer, (c) *The status of optical signal processing*, to propose new applications of optical processing, (d) *A review of the state-of-the-art in communication system design*, to describe the critical problems presently confronting designers, and (e) *A group discussion: the role of government, industry, and universities in advanced communication system engineering*, to bring out the cooperative and managerial problems that must be solved to make real improvements in the communication system design process. Each panel consisted of formal presentations and discussions.

All sessions were tape-recorded and transcribed in an effort to accurately preserve the sense of the discussions. Transcripts of presentations were edited for clarity and for inserting references to the speakers' slides, and the results were approved by the presenters. Hence the following proceedings are, for the most part, not formally prepared papers, but edited transcriptions.

Special thanks for the timely and high-quality production of these proceedings go to Cathy Cassells and Milly Montenegro, the workshop secretary and administrative assistant, respectively. Thanks for a job well done also go to Nikos Pronios and Gregory Yovanof, who supervised the recording process, and edited the transcriptions of the sessions.

Dr. Robert A. Scholtz
Workshop Chairman

USC-CSI WORKSHOP ON ADVANCED COMMUNICATION SYSTEM ENGINEERING

Proceedings of Session One: Design Tools of the Future

SCHOLTZ: We're going to record everything that's said here, transcribe it, and get a full proceedings of the meeting. So you'll see microphones around on the front tables -- I hope the speakers will pass them around as we answer questions. We have one roving mike and one of the fellows will get it to anyone in the audience who has questions.

The interest that USC has in this is trying to decide what our communications engineering group should be doing in the next 10 or 15 years. We'd like to know where universities fit into this process, both in teaching and in design, and what kind of research would be most useful to people in industry. I hope that in the next couple of days, we can answer some of those questions or at least get some good ideas about them. At this time, I'd like to turn this over to our speculative panel, run by Dick Booton, trying to "crystal ball" what should be happening in the future in the world of design tools for communications engineering. Dick ....

BOOTON: Thank you, Bob.

Bob asked us to be the first panel. As CHART 1 shows, it is assumed that we're *talking about design tools for the future*, that you already know what the present tools are. [laughter] But if that doesn't work, you're going to hear this afternoon about some of the present tools. I've been wondering about this microphone ... apparently it picks up radio signals too. We thought this morning we'd have six speakers.

We want to say a few words about what we see about the design tools of the future, and we've deliberately tried not to make the talks tightly integrated. You're going to hear different points of view on different subjects. Here's a general comment: We're going to try to talk about design tools of the future, both what kind of design tools do users want for the future (and this is presumably improvements in current tools if you know what those are) and new types of tools. And then, from an opposite point of view, what kind of design tools are likely to be developed in the near future. Hopefully we'll have some speculations about the far future. As I said, we assume that you're going to hear something about current design tools in other sessions.

As shown in CHART 2, the speakers this morning are Dale Harris, from Pacific Bell, who is going to talk more about networks; Ken Porter from Motorola; Barney Reiffen from Lincoln Labs, MIT; Raul Rey from TRW; Jim Spilker from Stanford Telecommunications; and George Turin from UC-Berkeley. Now I caution people to talk not about current design tools, but about the future. Raul Rey will probably say a few words about BOSS (Block Oriented Systems Simulator), anyway. [laughter] But I warn him I'll drag him off the stage if he says too much about it.

Some overview comments on what design tools are as outlined in CHART 3, and first of course we will talk about the analytical results that most university professors worry about. The detailed analysis made up from noise theory, modulation theory, and so on. Then there is a whole

set of numerical results which have already been obtained and are useful as background tools. More and more interest is in the question of signal simulators which is a matter of both hardware and software. As I said, one of these is what we've been working on together with Kansas University is the BOSS, which you'll hear more about later. Technical performance monitors, which are primarily software tools, will be doing the accounting in complex systems, keeping track of dB's and so on. Then there are databases that describe the components of subsystems that are becoming more and more a part of com-system engineering. Probably in a sense further in the future, but specially important, are software for processing test results and correlating those with the design tools, and with current designs.

One last CHART [4] of what I see as some of the needs for the future. First of all, much more detailed and realistic analyses. For years people have done analysis of simplified problems that were simplified enough so you could do a neat solution, but in fact the real system was always much more complex. There's need to find the analysis of realistic models. In the design of a real system one of the most important things that the com-system engineer does is to prepare specifications. That used to be a purely manual exercise and is becoming more and more of a computerized task that is closely tied in with the design of the system.

There are a couple of points of the interaction of the design tools. One is with circuit and subsystem CAD -- CAD in the sense of circuit design and PC board design and so on. It used to be a completely separate exercise. The com-system engineer would do his analysis and then somebody else, after some manual interaction, would worry about the circuit design, and then somebody would do the body design, and so on. These have always been, up to the very recent past, completely separate exercises with people and pieces of paper being the connection. As more and more of these things are handled electronically, there is more and more pressure to try to make these compatible. I think this will come out in some of the talks. The tying together of the system specifications and test results, and the final evaluation of the system as to whether it meets performance tends to become automated. A lot of the design tools require a very bright system engineer (or a team of system engineers) in the middle running the tools and working on the overall system design.

There's been a lot of loose speculation about Artificial Intelligence and how much one could put some of the creative design work in the hands of a piece of software. There was mention of talking about the artificial system engineer. This may be a long ways in the future but it's worth speculating. I'd be interested in comments from the group on that.

All of this talk about doing a better job of system engineering and more thoroughly integrating the com-system engineer's job into the rest of the design means, I think, that the job of what used to be called a system engineer is becoming much broader. I'd be interested in comments, either by the panel or by anyone in the audience, about what is happening to the job of the com-system engineer. Because I think it's changing, and in a way much more toward the system engineer being a real system engineer and in control of the overall engineering of a system, instead of simply being an analyst off in a corner doing some dB calculations. That's all I want to say for now -- let me come

TRW

Design Tools of the Future

Richard C. Booton, Jr., TRW

27 May 1987

CHART 1

Purpose of this Session

TRW

- This session will cover
  - What design tools do users want for the future?
    - Improvements of current tools
    - New types of tools
  - What tools are likely to be developed in the near future and in the far future?
- Current design tools will be covered in other sessions

CHART 1

366D2 587

Speakers for this Session

TRW

- Dale A. Harris, Pacific Bell
- Ken Porter, Motorola
- Barney Reiffen, Lincoln Labs
- Raul Rey, TRW
- Jim Spilker, Stanford Telecommunications
- George Turin, UC Berkeley

CHART 2

36603 587

TRW

Design Tools for Communication Systems Engineering

- Analytical results
- Numerical results
- Signal simulators – hardware and software
- Technical performance monitors
- Component and subsystem data bases
- Software for processing test results

CHART 3

36604 587

Needs for the Future

- More detailed and realistic analyses
- Better tools for specification preparation
- Interaction of system design tools with circuit and subsystem CAD
- Interaction with data bases describing subsystems and components
- Comparison of system specifications and test results
- Introduction of artificial intelligence?
- Better understanding of the changing role of the systems engineer

CHART 4

36605 587

back later.

Let's go back to the speakers and since there's no particular reason for convening in any order, I'd like to ask Dale Harris to start. We're going to try to hold these to approximately 15-20 minute talks, and 6 times 20 is two hours, so depending on how much interruption and questions we have ....

HARRIS: My topic today is the design of networks, and in particular the design tools of the future that will help Pacific Bell appropriately design the public network, as it's technologically evolves.

At Pacific Bell, we spend almost $2 billion a year to economically upgrade and evolve the public network. Someone once said, while talking about the defense budget, "A billion here and a billion there, and pretty soon it adds up to real bucks." There are seven companies in the U.S. similar to Pacific Bell, so that's $14 billion, and that's just for the local exchange networks. A major challenge in economically upgrading and engineering the network is deciding what switches and other facilities to deploy and where to deploy them. We need to design the system so that it functions efficiently and meets customer needs.

In looking to the future, there will be many technological changes -- changes which will impact network design. For example, the public networks are moving from the analog system of today to a fully digital system. One of the design problems this change poses is that we can't simply remove everything that's out there and start over; instead, this change will occur as an evolutionary process. Currently, we are placing digital facilities in the network, but it will take several decades before the entire network becomes fully digital.

Today, we provide "discrete services"; that is, you have a connection in your home for each service you receive. For your telephone or personal computer, you have a telephone connection. For your television, you have a separate cable into your home. In the future, we may provide "integrated services," where multiple types of services -- data, voice, video -- will be offered over the same facilities.

Another change that's going to be important is the use of fiber optics in the network. Many of the facilities between central offices today are fiber optics. In the future, we will begin to see where economic, fiber optics in the "loop plant" are the connection from a central office to the home or business. The fiber optic loop may begin to become prevalent in new construction in the mid '90s.

A fiber optic local loop will impact our current approach to network design. Today we have a fully interconnected, highly meshed network architecture. When we begin to deploy very high capacity links for the fiber optic loop, the architecture becomes thinner. This may result in more possibilities of a failed link causing total network failure. Thus, the economic tradeoffs between lost revenue and survivability and cost will become a more important dimension of network design.

These technological changes -- the move from analog to digital, from discrete to integrated services, and from copper to fiber in the local loop -- will change the criteria used to design the network (FIGURE 1). In the past, our design criteria have included quality of service, and number and length of calls. In the future additional criteria will be used including error rates, throughput, and transport delay (FIGURE 2). In most cases, the design

tools that we use today may be made to accommodate these types of changes by shifting the design parameters.

I believe that the network trend most likely to drive changes in our planning tools is the trend towards a real-time environment, a network that responds almost instantaneously to a request for change. Currently, our network is not reconfigured in real-time. Changes in the network are generally done by an order entry process, using human and batch processing. We design physical circuits; we set up physical connections between points. In a real-time situation, a customer will be able to order instantaneous changes in service. In the case of the large customer who has a large number of facilities, such changes would constitute a reconfiguring of the network on a real-time basis. We will do this using logical circuits; there will not necessarily be an identifiable, physical connection for each point-to-point connection. This approach will require significant use of multiplexing and packet networking. FIGURE 3 summarizes the changes that will occur as a result of a real-time environment.

Today two types of tools are in use: design tools and management tools. Management tools are computer-aided tools used in the day-to-day monitoring of the network and in insuring its appropriate functioning. Design tools are used to aid in decision such as the installation and location of facilities. All the design is done off-line, rather than in real-time.

Public networks are very, very large and complex. Today we have multiple databases and management systems to manage them. The network management and the network design functions are separate but use much of the same data and many of the same databases. In the future, network management and design will trend toward a single system.

Currently, two types of computer aided design tools commonly used are calculator tools and logical search tools. The calculator type of tool is used by the engineer to perform numerical analysis. It provides both economic and technical results. However, the design engineer makes the decisions using the calculator tool as a specially programmed calculator.

A logical search tool allows the engineer to specify a number of alternative solutions and the computer numerically assesses them, after accessing data from appropriate databases. The computer derives the optimal solution to the problem given the alternatives presented by the design engineer. One problem with logical search tools, as we look to the future, is that the run times are very long. In fact, the network is so large that an optimization run for a design problem, using a geographical area such as Los Angeles, may require hours of computer time. So this approach cannot be adapted to real-time requirements. In practice, the calculator tools give mixed results; the run time and accuracy can vary.

What about using expert or knowledge-based systems for design tools? FIGURE 4 shows a comparison in performance of calculator tools, logical search tools, and expert systems. According to FIGURE 4, expert systems potentially may out-perform both calculator and logical search tools. Several expert systems have been built for network maintenance and management. I will describe seven of these expert systems. Five of these are prototypes and two are being used in the field to aid in network design and management.

FIGURE 5 shows these seven expert systems.

Designet is a knowledge-based system by BBN Communications Corporation in Massachusetts. The design engineer provides network requirements and asks the system to generate a first guess of what an appropriate design might be. The design engineer can then refine the design, change various characteristics, and run simulations on the system. Presently, a programmer is required to update the knowledge base. Future goals of Designet are additions of an easier knowledge base interface and migration of the design process into network management as design becomes more real-time. Designet is used internally at BBN to design networks. It has not, to my knowledge, become a product for sale.

Net/Advisor is an expert system developed by Avant-Gard Computing, Inc. It works in conjunction with other network design and management tools. It provides knowledge base access to a number of traditional network design and management tools, such as the IBM network communications control system, for example. The system does exactly what its name implies -- it gives advice to the designer at the workstation. The system itself does not make decisions; rather it offers advice to the designer and the designer makes the appropriate decisions. FIGURE 6 shows the key components of this system. Net/Advisor is a knowledge based enhancement to a conventional product, Net/Command. In future revisions, Net/Advisor may become an active network element and pass commands to network devices as well as recommendations to the designer.

Bell Communications Research has developed an expert system, SMART (Switching Maintenance and Analysis Repair Tool). SMART was designed for the purpose of replacing human experts and providing an automated capability for maintaining the switches in the central offices. FIGURE 7 shows how SMART is designed to work with switching components. Presently, it works only with AT&T switches.

Nemesys (Network Management Expert System) and ACE (Automated Cable Expert) are expert systems designed by AT&T. The purpose of Nemesys, which is in the prototype stage, is to optimize network performance during periods of overload and stress. The ACE system is currently in use in the field; it helps the engineer identify cable problems and helps design solutions to those problems. Cabling problems are a useful application of expert systems because these problems vary in different parts of the country. For example, different parts of the country vary with respect to weather conditions and cabling is affected by weather conditions. By using local experts to develop the approach to cabling problems and cabling designs, the expert system is able to capture some of the localized expertise in an automated system. Eventually, the goal of ACE is to aid in the long-term design of transmission facilities; this system is in place in four companies.

Compass, a GTE Laboratories system, is an expert system aimed at the design and maintenance of switching systems. NTC, Network Troubleshooting Consultant, is a DEC expert system. It functions in DEC's remote maintenance centers. When a DECNET customer calls a problem in to the remote trouble center, the NTC expert system analyzes the symptoms of the trouble, makes a decision about remote diagnostics that ought to be run, then calls (through another software package) the network

TECHNOLOGY CHANGES AFFECTING NETWORK DESIGN

Present		**Future**
Analog	→	Digital
Discrete Services	→	Integrated Services
Copper Cable	→	Fiber Optics
Voiceband Services	→	Broadband Services (Video, Data, Voice)

Fig. 1

PRESENT AND FUTURE CRITERIA FOR NETWORK PERFORMANCE

Present		**Future**
Quality of Service Call-Seconds	→	Error Rate Through Put Delay

Fig. 2

NETWORK CHANGES DUE TO A REAL-TIME ENVIRONMENT

Present	**Future**
Batch →	Real-Time
Order Entry →	"Instant Service"
Physical Circuits →	Logical Circuits

Fig. 3

A COMPARISON OF CALCULATOR, LOGICAL SEARCH TOOLS, AND EXPERT SYSTEMS

	"Calculator"	Logical Search	Expert System
Result	?	Optimal	Good
Flexibility	Good	Poor	Good
Application	Fair	Limited	Wide
Runtime	?	Long	Short
Ease to Build	Easy	?	Hard
Developed by	Industry	Universities	?

Fig. 4

EXPERT SYSTEMS

Designet

Net/Adviser

Smart

Nemesys

ACE

Compass

NTC

Fig. 5

NET / ADVISER

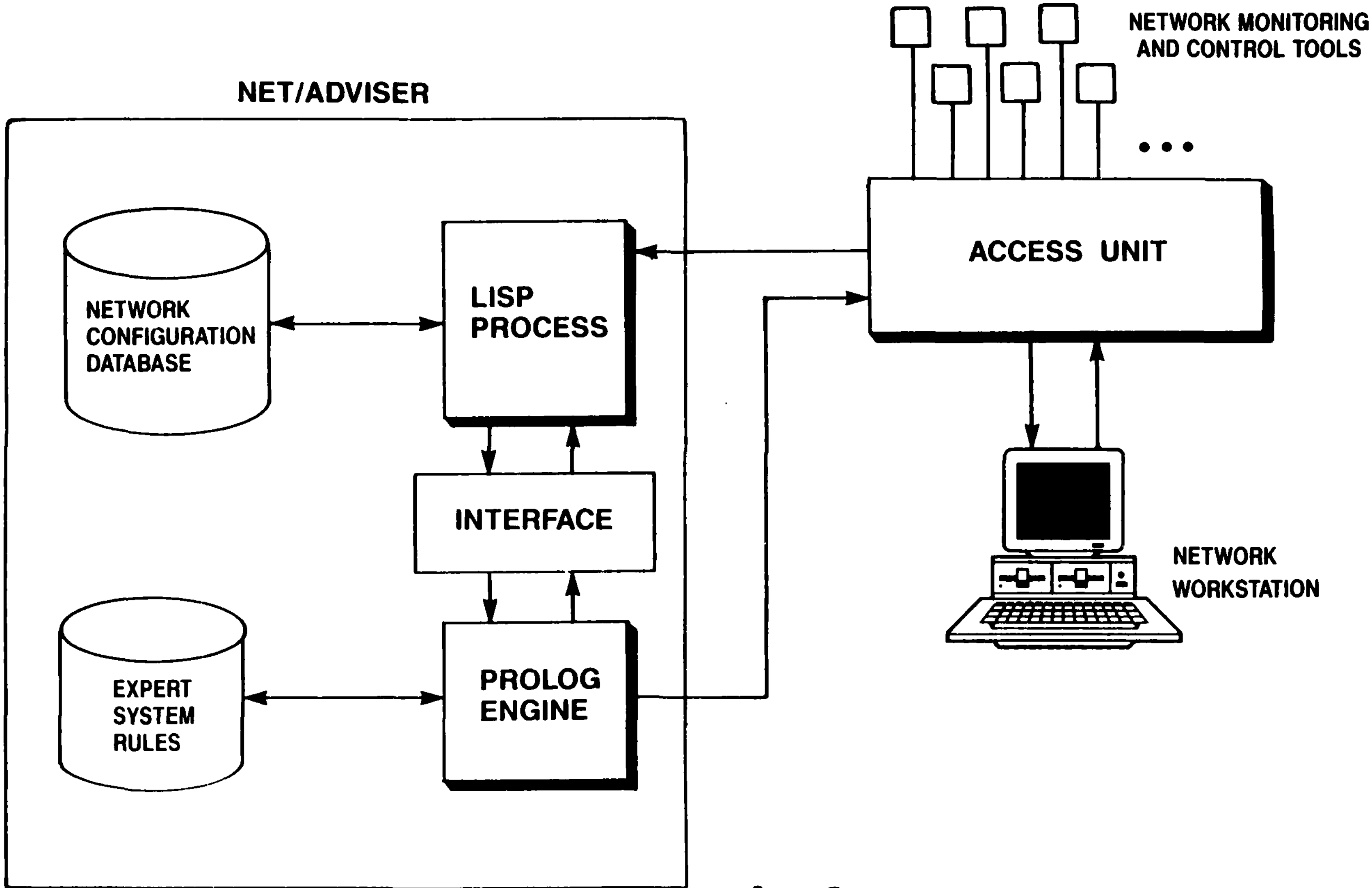
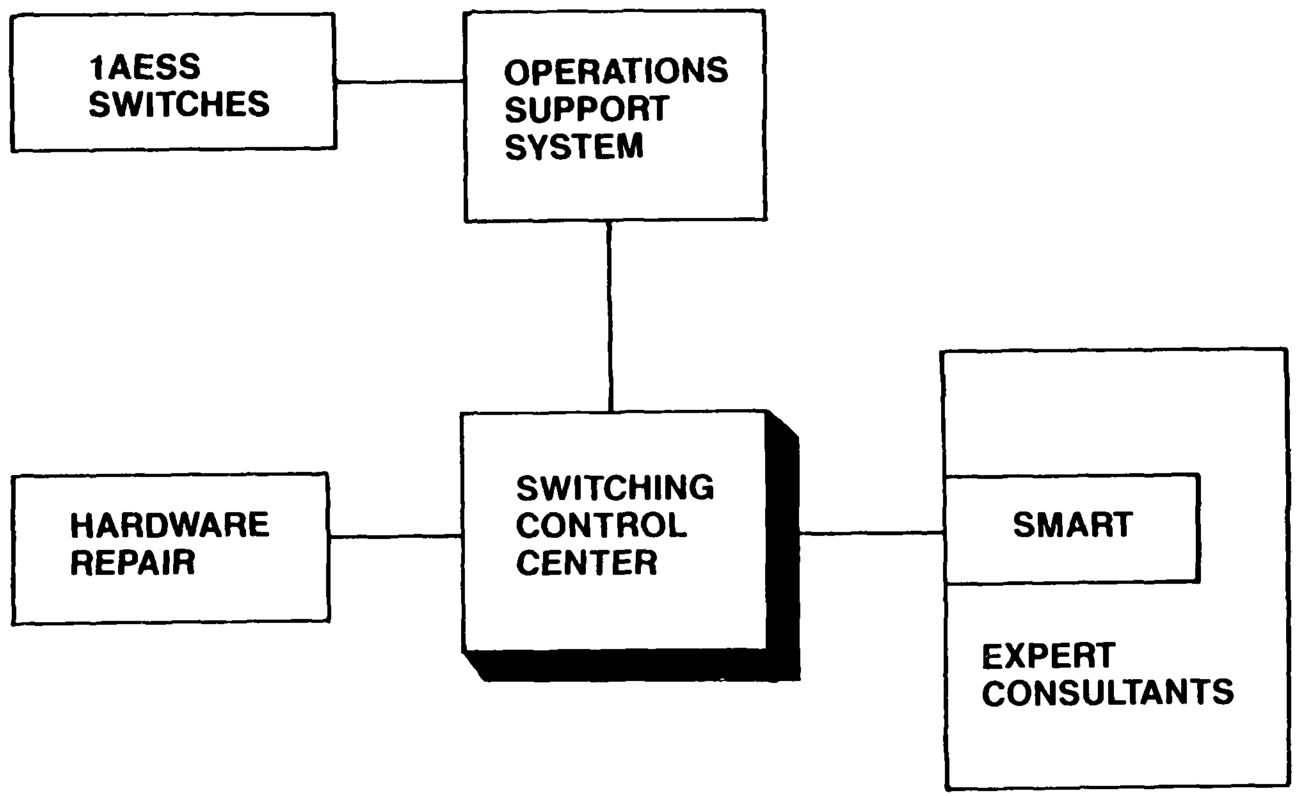


Fig. 6

SMART



1AESS SWITCHES
OPERATIONS SUPPORT SYSTEM
HARDWARE REPAIR
SWITCHING CONTROL CENTER
SMART
EXPERT CONSULTANTS

Fig. 7

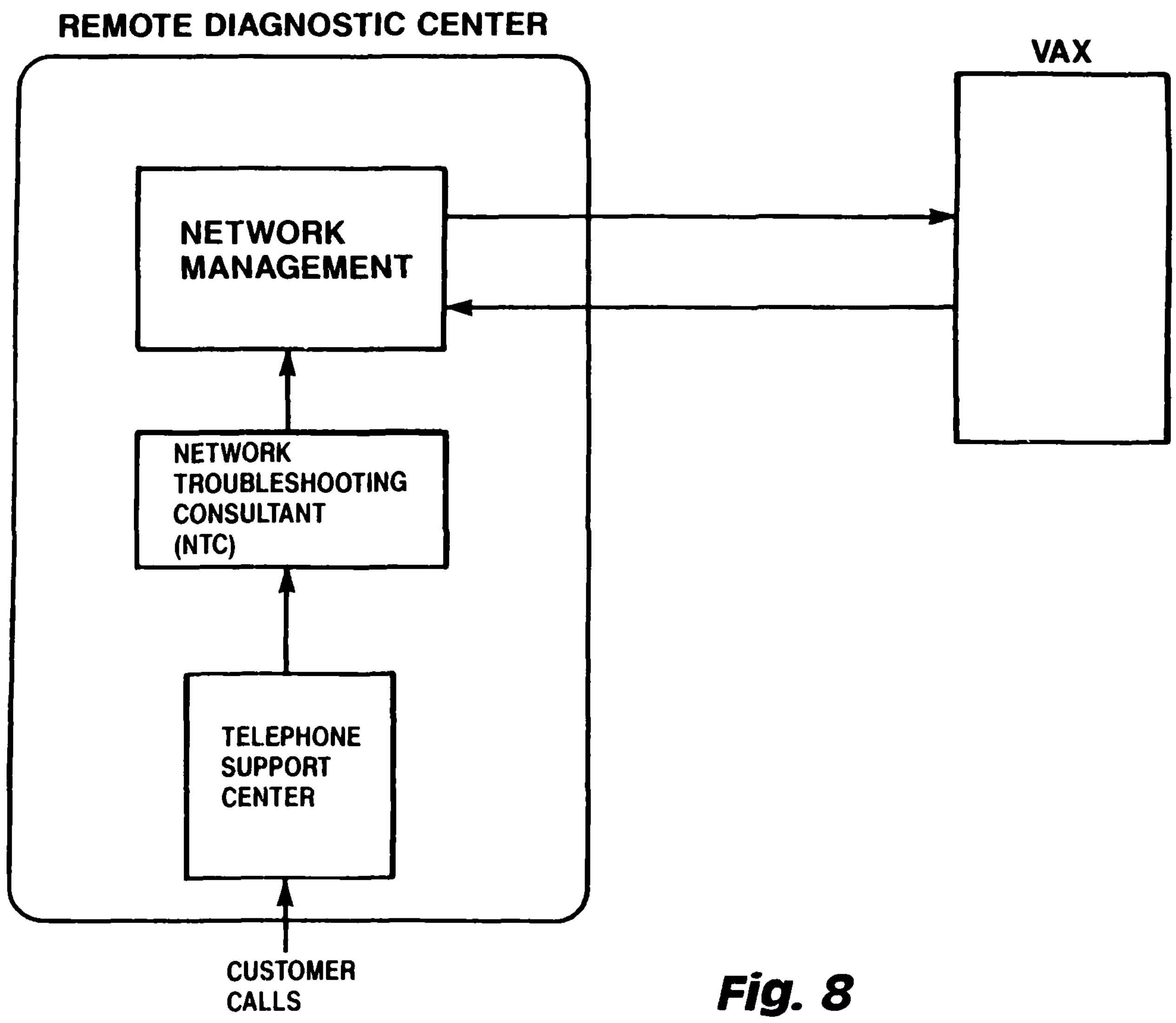

REMOTE DIAGNOSTIC CENTER
NETWORK MANAGEMENT
NETWORK TROUBLESHOOTING CONSULTANT (NTC)
TELEPHONE SUPPORT CENTER
CUSTOMER CALLS
VAX

Fig. 8

manager and runs those network diagnostics remotely. FIGURE 8 shows the components of the NTC system. NTC is not a commercial product, partly because of the proprietary knowledge embedded in the knowledge base.

Today, most expert system tools are being developed in the area of maintenance and management. But as we move to the future, the distinction between design of the network and management of the network may become smaller and smaller. Several of these prototype expert systems have the goal of becoming near real-time systems and allowing the user to not only diagnose network problems but (through some additional software) to design network changes as well. These types of intelligent, real-time tools are necessary to achieve the ultimate goals of network design and operation.

MOHANTY: Are there systems for data networks or for voice networks?

HARRIS: They will be for both.

MOHANTY: Both?

HARRIS: For both. A lot of the tools being developed today are being applied to data networks, for example the DEC NTC application. But, as I mentioned, in the future we're going to see voice and data integration, so that instead of having separate networks for data and voice, there will be a single integrated network.

MOHANTY: Thank you.

HUTH: You talk about short run times for expert systems, and I'm surprised -- I guess I don't understand why you think they would have short run times versus having a human expert sitting with a computer. He's played the system for 10 or 15 years. Why does it take longer for him to do the job than it would be for an expert system?

HARRIS: Today our network design tools for the public network can take hours to run due to the network size and complexity. They are run in batch mode on top of that. It takes a long time to make a design decision.

With an expert system, the approach is not to try to achieve the optimal solution but a good enough solution. The run time is a lot faster and eventually design decisions can be automated so that there really isn't even a human designer involved at all. For example, if you look at the DEC system, the expert system ideally makes the decisions about what diagnostics to be run and runs them. Ideally, there is no need for a human being to be involved at all. Some of the other systems I described today are trending toward having the expert system arrive at decisions and develop the detailed configuration specifications for the design. In the case of the network of the future which will rely less on physical circuits and more on virtual circuits, the network can be redesigned in a virtual sense in near real-time.

SANDER: In the beginning you mentioned that you saw that commercial and public networks will use packet technology, and at the same time you indicated that you expected to see an instant service kind of performance in the future. To my way of thinking that's somewhat incompatible because it takes finite time to get packets through the network. Do you have any comments?

HARRIS: Well, what is meant by instant service is that a change in a customer's required services or facilities may not be done by calling up the telephone company which in turn generates a work order. Rather the changes are implemented through automation. To achieve the

flexibility required, virtual circuits are preferred. As you know, virtual circuits are an integral part of packet technology. Any delays caused by packet processing will not be detectable by the end user if the system is engineered correctly.

SANDER: Thanks.

BOOTON: I think we ought to move on along -- I know there are some more questions -- but our aim was to have some time at the end for more questions with the panel and some panel discussion. So Ken Porter from Government Electronics, Motorola .... Save your questions for a day later.

PORTER: Do you want them to entertain questions as we go through the presentation?

BOOTON: I don't mind short questions here, but don't involve a real discussion -- let's leave that for the end.

PORTER: Thank you. What I'd like to do is define system engineering a little bit broader than you are accustomed to thinking and discuss some of the tools of the future. The Motorola Government Electronics Group, is a contract development and production organization for DoD. We must be very efficient in system engineering and design. Half of our business of contract engineering, developing products that later on we will build or will be manufactured by other contractors. For the last 3 or 4 years, we've been involved in designing a computer integrated design environment for our DoD business, where we can take customers' requirements and flow them down to detail design descriptions, do the physical design of the hardware, and build the product. We've been in the process of evaluating software tools for this entire design process. What I'm going to discuss today is what I believe are some of the key factors driving the tool requirements, (particularly in the system engineering portion), but also the status of the software tools themselves. We've been evaluating literally hundreds of software tools. They all have one thing in common: they don't work together and that's one of the big problems we have. We're looking for an integrated design environment. We do not believe that we can economically develop these tools ourselves. Furthermore, we don't believe that we can effectively integrate software tools together at a low level and then maintain that system. We therefore have been careful to pick software tools where large portions of them are already integrated together.

(VIEWGRAPH #1) This is our overall goal, and it's an ambitious undertaking.

If you look at what's been happening in engineering, the development times have been increasing over the years instead of going down. One of the reasons for this increase is that the time required to define the system has been going up, but some progress has been made in the hardware design. So we see the actual development time increasing instead of decreasing. We need to turn this trend around because if you can reduce the development time, you also reduce the development cost -- *it's just* that simple. The second trend is that the producibility of the products as they reach the factory floor is also going down because of increasing complexity. I want to address these issues; system design and system complexity. I have selected some examples of communication equipments which you are familiar with, to discuss here, in order to show complexity trends in some of these equipments. These equipments are relatively small physically but very complex functionally. What truly is driving system engineering is the fact that

the design technology is going largely to digital techniques and the ability to design VLSI circuits is increasing rapidly; therefore the difficulty of defining these complex equipments is increasing rapidly.

I will describe 4 technology charts which I call "Complexity vs. Time" and the technology is VLSI products, pager products, software and military narrowband digital voice system. These are typical VLSI products. (VIEWGRAPH #2) Most of them are familiar like the Motorola 6800 and the 68,020. You will notice, as would be expected and as people have talked about, the complexity of these chips; they are increasing by a factor of 10 every 5 years. Also, as you would expect the memory chips like the 64K dynamic rams are more complex than the general trend because of the systematical design pattern.

(VIEWGRAPH #3) Now with the availability of every increasing VLSI technology as standard products and custom chips, the current communication systems that are built, are about half standard product and about half custom VLSI. To define the requirements for these communications systems and the interaction of these complex chips is becoming an increasingly difficult problem which is generally multiplying faster than our ability to actually design VLSI chips.

(VIEWGRAPH #4) The next chart shows the complexity vs. time when you apply this high technology to an electronic ballpoint pen, communication pager. Not surprisingly, even in a small product like a communication pager we're now seeing a complexity of over 10,000 equivalent gates which includes a complete receiver, filter and crystal frequency control -- and data processor that stores messages. The devices are small enough to fit in your pocket but are becoming very complex because of the ability to integrate VLSI into them (VIEWGRAPH #5) Another complexity issue is the software required for generalized signal processing communications function. Signal processing software complexity is also growing at a very high rate. This chart for some Motorola communication products plots the number of software CPU instructions that are used in the communication equipment versus time. Again you see an exponential growth rate of more than 10 to 1 in less than 10 years. These equipments are base-station equipments that control the cellular and trunking communication centers. All of these complexity factors; VLSI, equipment functions and software are compounding together. A good example of a product that illustrates the impact of complexity compounded is shown here. (VIEWGRAPH #5) These are narrow band digital voice equipments for the U.S. Government communication DoD market. They've gone to digital communications in order to provide security. Shown here are the million instructions per second that are executed by these equipments to provide communication and security. Again, you can see the ever-increasing complexity to achieve the performance needed. I'll talk to the complexity of the STU III system a little later, but let me point out that it executes nearly 10 million instructions per second in order to transmit a high quality secure voice message. But, how does all this increasing complexity apply to system engineering?

(VIEWGRAPH #6) Illustrated here is a very simplistic system engineering design process chart shown here in order to orient our thoughts so that we can put some structure into a classical system engineering task. The input into the system is customer requirements. These requirements need to

be analyzed and solutions synthesized for them. These solutions must be broken down into some form of modularity and allocated to each subsystem which then must be further defined by a preliminary and detailed design. The important point is all of the complexity trends that I talked about on the previous charts are occurring down at this lower level; subassembly, modules, and component design. Our ability to design and build components like microprocessors, subassemblies and black boxes, is increasing much faster than our ability to basically define the requirements and simulate these functions. The other factor that is a major constraint is that the completed system during the design verification and integration phase are so complex, that it is almost impossible to test and validate the entire system design. We are finding that there's no systematic way for the system engineer to manage the process of defining requirements and making sure that the final design meets the preliminary design requirements and making sure that the final design meets the preliminary design requirements and validated at all these steps. Therefore, our ability to design hardware, software, and VLSI circuits is outgrowing our ability to define them and to test them. That's what we're finding. Our approach is to create a design environment for system engineering that will manage this process by putting a structure into process, so that it can be systematic and provide the automated tools. The system engineer can then concentrate on the conceptual design, the analytical process, as opposed to trying to keep track of all the requirements manually.

(VIEWGRAPH #7) Shown here is a classical system engineering program phasing; system engineering, preliminary design, detail design, integration and tests and qualification and test, together with an overlaying systematic review process.

(VIEWGRAPH #8) Adding the system design disciplines to this overall program phasing provides a framework in which to discuss the maturity of software tools.

The areas that I discussed earlier where we have the most advanced capability today is in the preliminary and detailed design phase. The first phase we call system engineering although system engineering goes across the entire process and is the glue that ties the entire design process together. System engineering also includes integration and test validation.

Software tools are being evaluated for each one of the phases of the process including system design, digital, analog, RF microwave, software design, the physical design, mechanical and PWB, and so on and so forth, including integration and tests. Before I discuss the software tools maturity I will show one last system design example of complexity. (VIEWGRAPH #9) The STU III is a DoD narrowband digital voice telephone which was developed for the U.S. Government for both conventional and cellular telephone applications. The voice quality is nearly the same as a typical telephone that costs $50.00. The unit in quantity will sell for around $3,000. It provides DoD-level security for DoD customers and U.S. Government personnel on a dial-up capability. It provides a capability that's not been available in the past. Primary difference is dial-up capability and good voice quality together with DoD security. Now let's examine what goes into this system, and consider some typical development times. Summarizing here is the software and hardware complexity of this equipment. It includes four 8-bit microprocessors, there's one 16-bit microprocessor,

two custom co-processors, four other digital processors, and one custom arithmetic unit. It also has standard logic of about 300,000 equivalent gates. The custom chips include 250,000 equivalent gates not including the memory; the operating system software uses 63,000 instructions and the test system has 50,000 instructions. This is a complex system to be implemented in a small conventional desk top telephone. But now the reason for describing this secure telephone. (VIEWGRAPH #10) The schedule for the STU III is shown along the bottom of this diagram. The development was completed in two phases. First the system design specification was completed in 14 months in 1985 and 1986. The definition of that secure network dates back, 3 or 4 years, and some of the technology for the linear predictive coding has been in development 15 years. We estimate if we had done this development 5 years ago in 1980, the telephone would be 10 times larger. It would have taken at least twice as long, perhaps three times as long. The time in the frong end design, the system definition, has not improved in 5 years. Very few new tools for system engineeering have developed in this period. What's happening is that with time for the actual detailed and physical design is decreasing but because of the complexity, the system design time is either remaining the same or actually increasing. The other perspective I'd like to share with you in summary, is our experience relative to what tools are available for the design disciplines and what is needed.

(VIEWGRAPH #11) Summarizing, in this chart is what we've found. At the bottom of the chart is a scale of 0 to 10. Nothing adequate means that there may be tools available but they're really not adequate. At the center position, a 5 means good tools but they're limited in terms of integrating with the other software system. A 10 means excellent tools that are integrated into the overall system engineering design architecture development. The area that is in the best shape, although, in my opinion, not totally adequate is software development tools. There are software development tools which provide a good frame work for defining the requirements, and the requirements analysis flow down, and then end up with actual compiled code. The next most mature software tools are the physical design such as mechanical and circuit board followed by the digital design. There's very little software for analog, RF and microwave -- just stand-alone software packages, nothing integrated together. There is a critical need to be able to do behavioral simulation of both a software and digital subsystems to tradeoff software and digital implementation but very little capability to do this exists. Probably the biggest problem that system engineering has is validating a design after it has been completed; before it is reduced to hardware or even after a system has been built. There is none other than a manual process for systematically comparing the results of system integration and test with the requirements. Also, if the design and implemented requirements start changing, and the customer typically changes requirements, flowing these new requirements down and making sure that all the changes are implemented in the hardware and is largely a manual process. Configuration management and a systematic way of flowing requirements through when changes are made is a critical need. Nothing like this in our opinion exists.

The other need which is even more serious is that behavioral simulators for the front design are developing with no common language. All of these tools use different languages. Also, none of these

GOALS FOR SYSTEM ENGINEERING AND DESIGN METHODOLOGY

PRODUCT DEVELOPMENT TIME –
2 To 1 Reduction In 5 Years
Requirements To Release To Production

QUALITY IMPROVEMENT –
100 To 1 Improvement In Design
Errors And Producibility

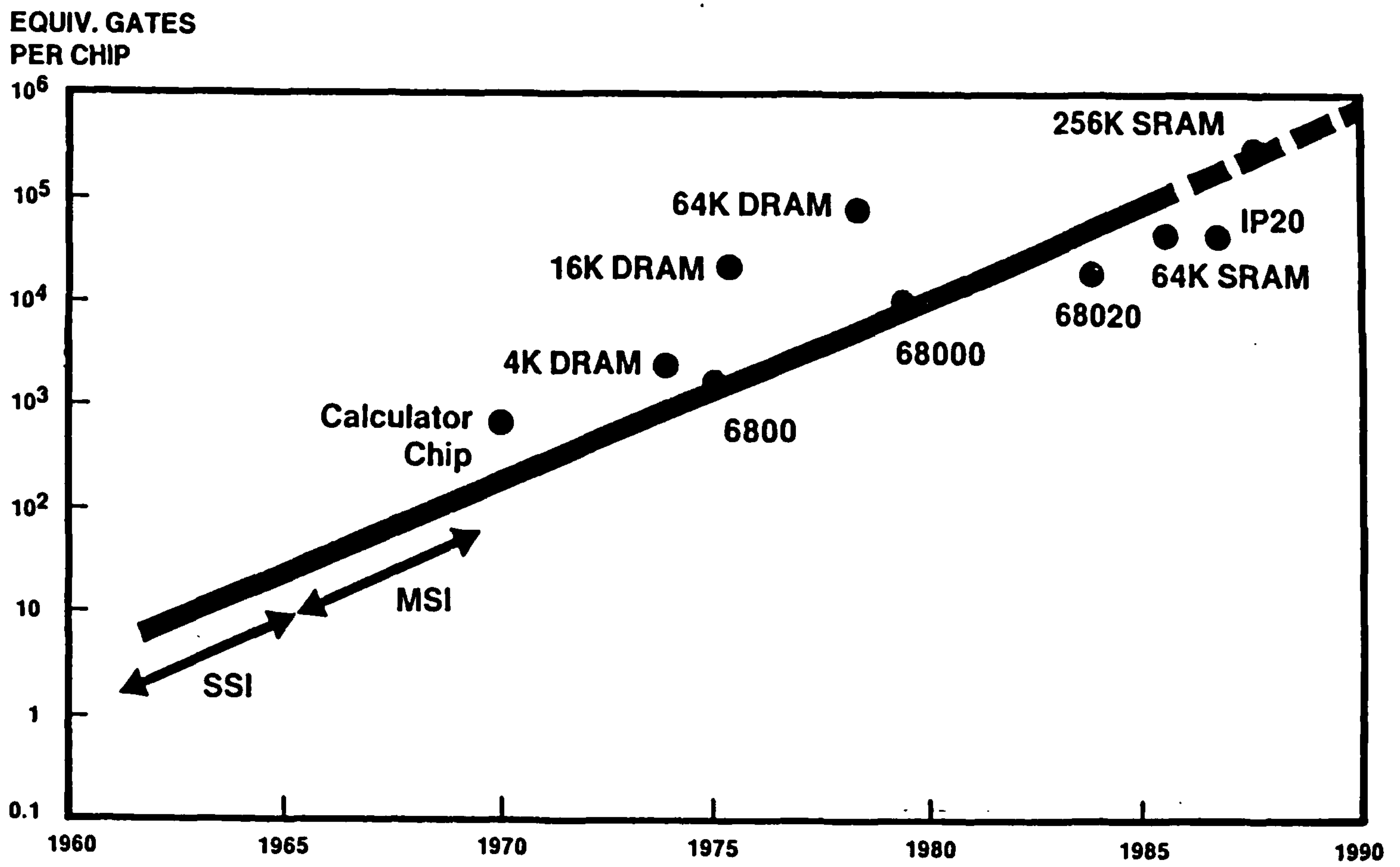
VIEWGRAPH #1

GEG

KEN PORTER (2-2)
MAY 27, 1987

COMPLEXITY vs. TIME

VLSI PRODUCTS



VIEWGRAPH #2

71974-2

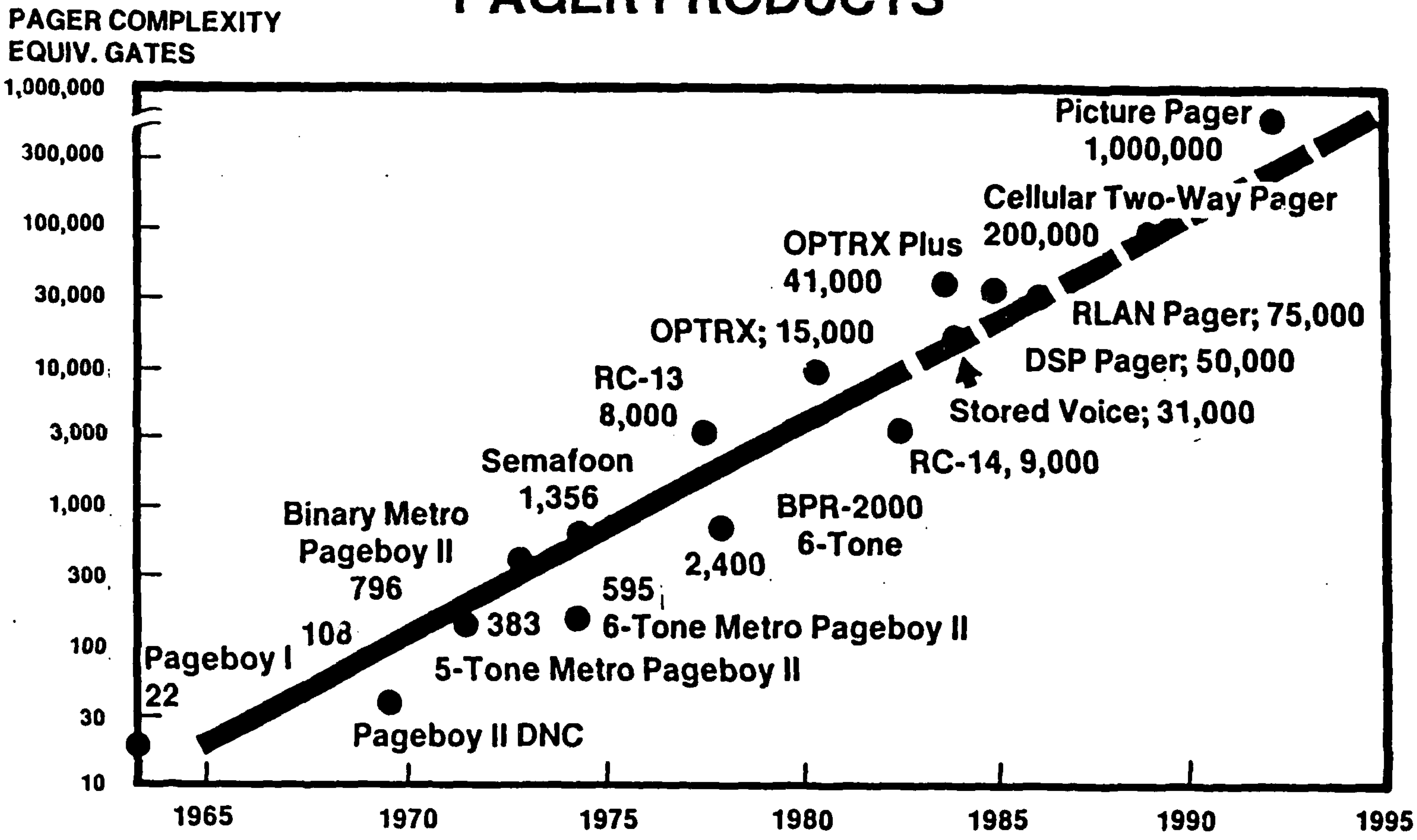
H2274

GEG

KEN PORTER (2-1)
MAY 27, 1987

COMPLEXITY vs. TIME

PAGER PRODUCTS



VIEWGRAPH #3

71874-3

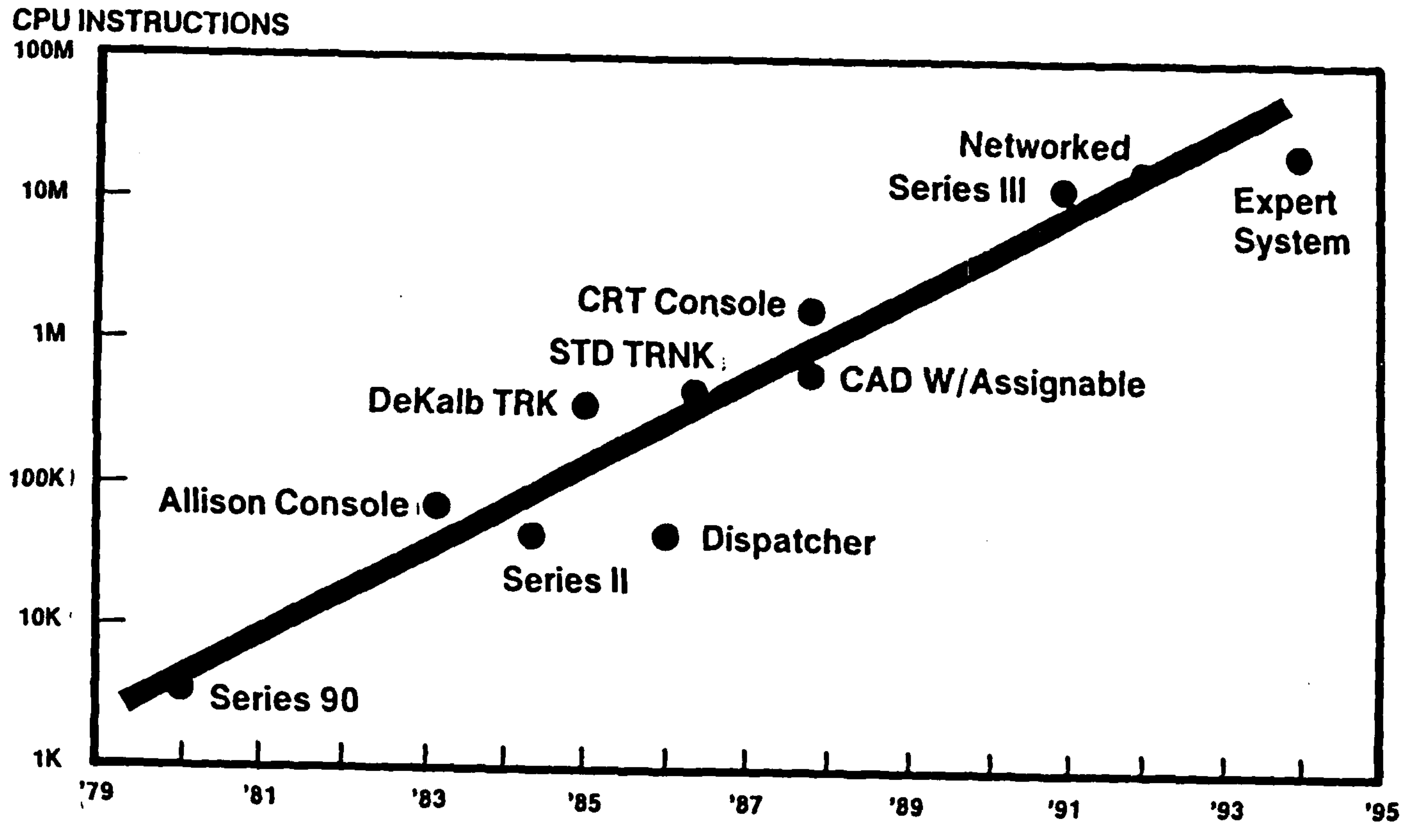
H2274

GEG

KEN PORTER (2-3)
MAY 27, 1987

COMPLEXITY vs. TIME

SOFTWARE FOR CONTROL CENTER PRODUCTS



VIEWGRAPH #4

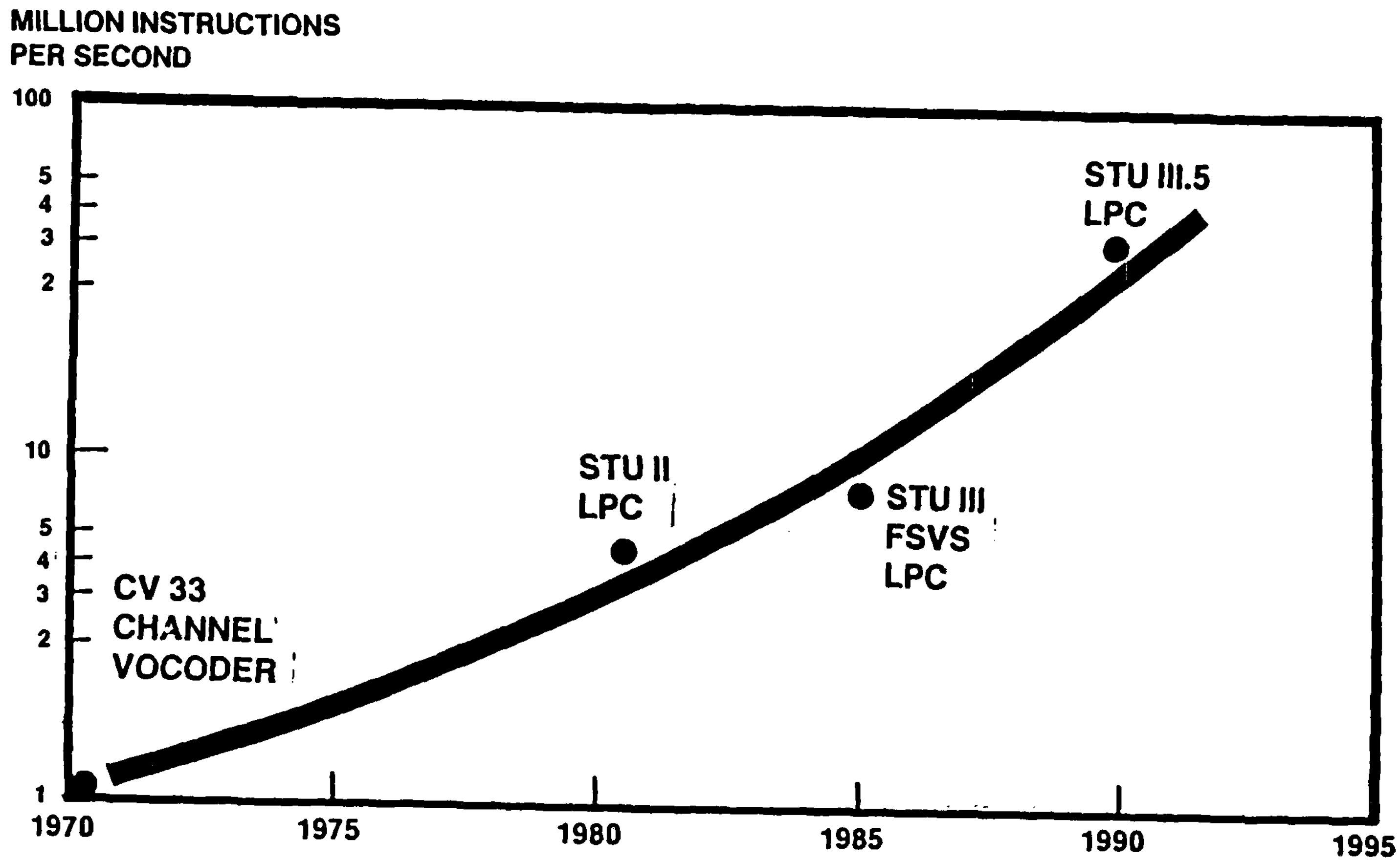
71874-4

H2274

KEN PORTER (2-4)
MAY 27, 1987

COMPUTATIONAL PROCESSING VS TIME

MILITARY NARROW BAND DIGITAL VOICE



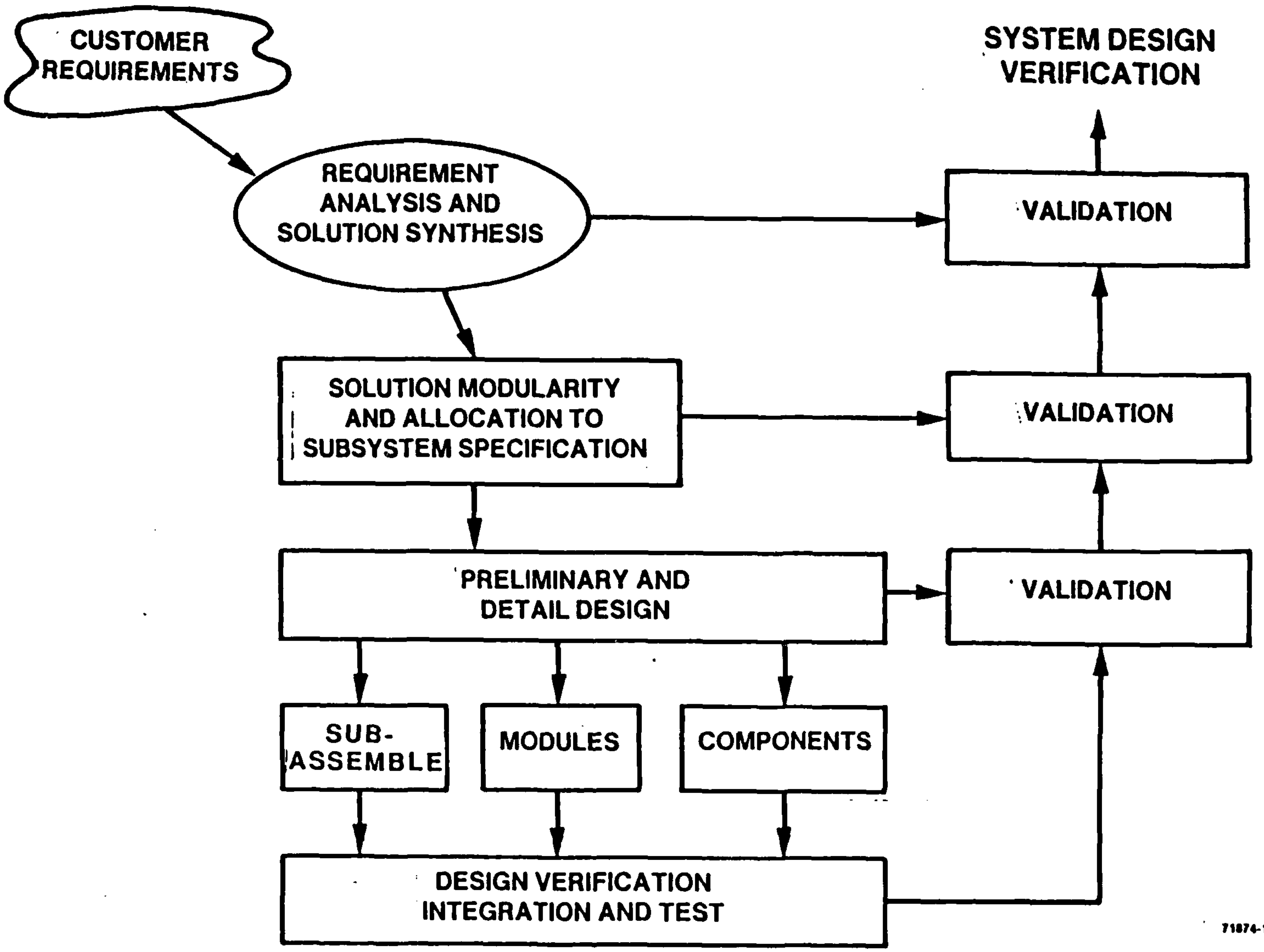
71674-5

VIEWGRAPH #5

H2274

KEN PORTER (1-1)
MAY 27, 1987

SYSTEM ENGINEERING DESIGN PROCESS



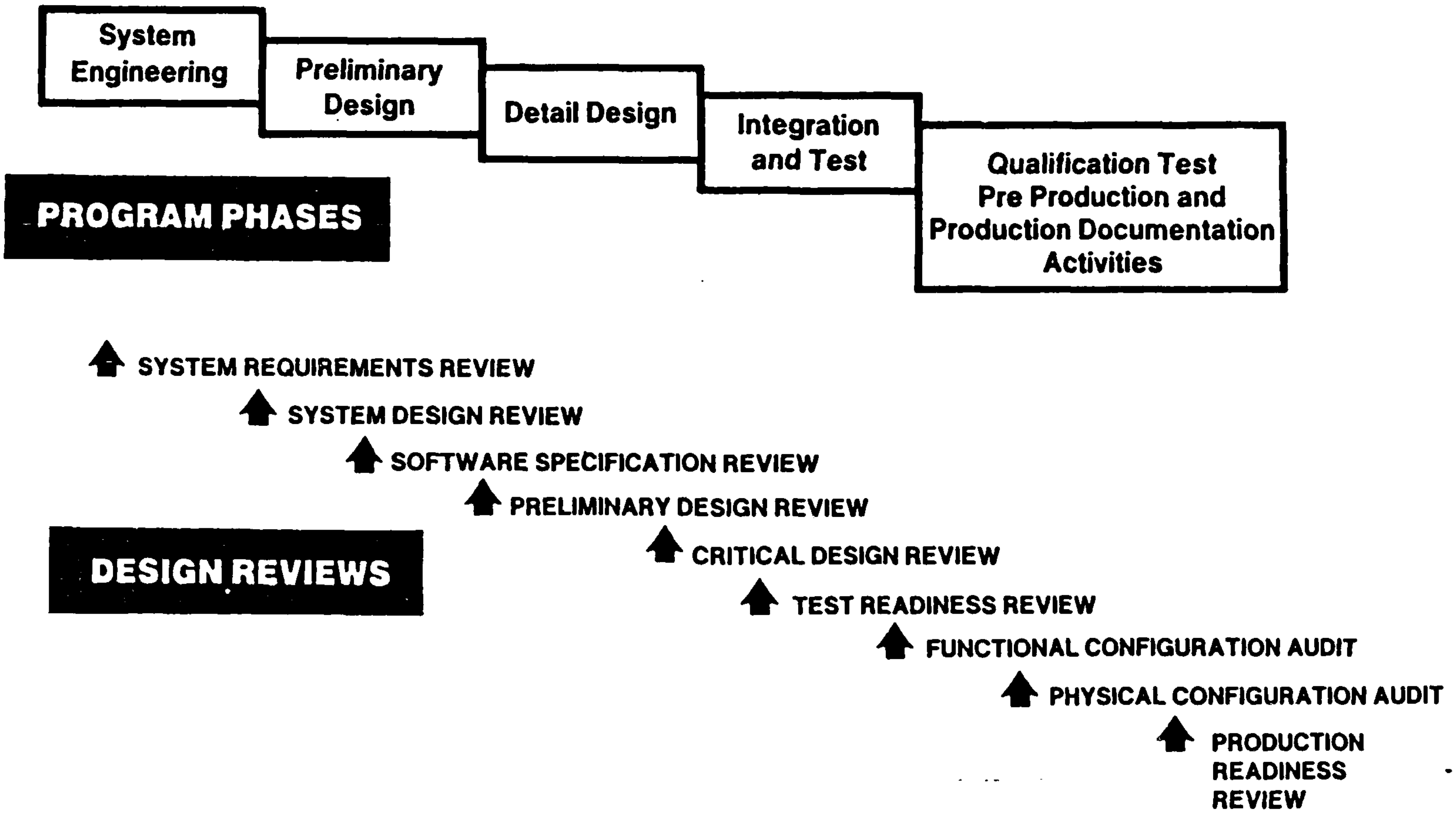
71874-1

VIEWGRAPH #6

H2274

KEN PORTER (3-0)
MAY 27, 1987

SYSTEM ENGINEERING DESIGN METHODOLOGY
PRODUCT DEVELOPMENT CYCLE AND REVIEWS



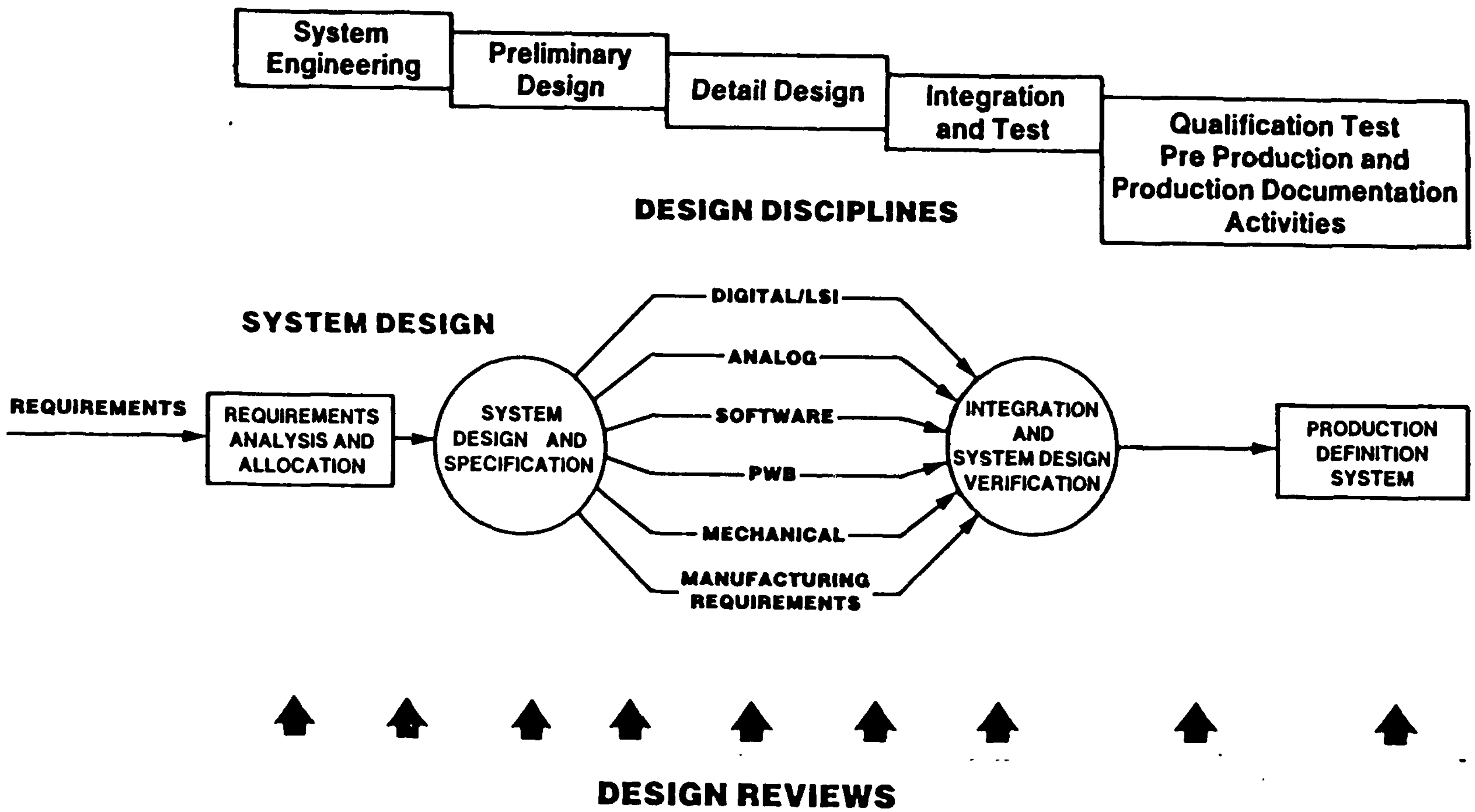
VIEWGRAPH #7

71874-10

H2274

KEN PORTER (3-1)
MAY 27, 1987

SYSTEM ENGINEERING DESIGN METHODOLOGY



VIEWGRAPH #8

71874-11

H2274

KEN PORTER (2-7)
MAY 27, 1987

STU III FSVS

SECURE TELEPHONE COMPLEXITY

HARDWARE

NUMBER	
4	8 BIT MPU's
1	16 BIT MPU's
2	CUSTOM COPROCESSOR
4	DIGITAL PROCESSORS
1	CUSTOM ARITHMETIC PROCESSOR

STANDARD LOGIC 300,000 EQUIV GATES

CUSTOM VLSI 250,000 EQUIV GATES
DOES NOT INCLUDE MEMORY

SOFTWARE CPU INSTRUCTION

STU III TELEPHONE 63,000
TEST SYSTEM 50,000

VIEWGRAPH #9

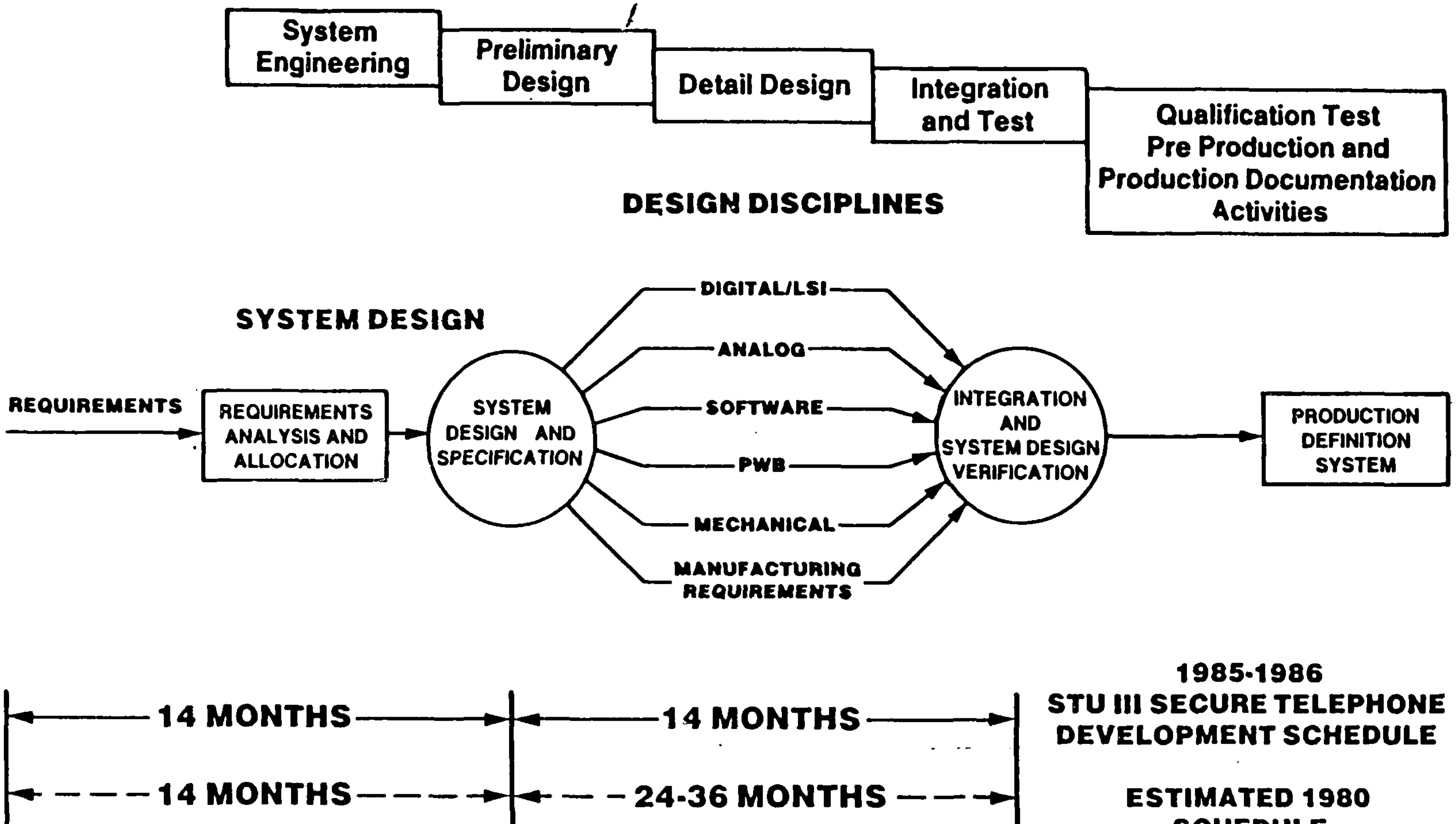
71874-8

H2274

GEG

KEN PORTER (3-2)
MAY 27, 1987

SYSTEM ENGINEERING DESIGN METHODOLOGY
PRODUCT DEVELOPMENT CYCLE



71874-12

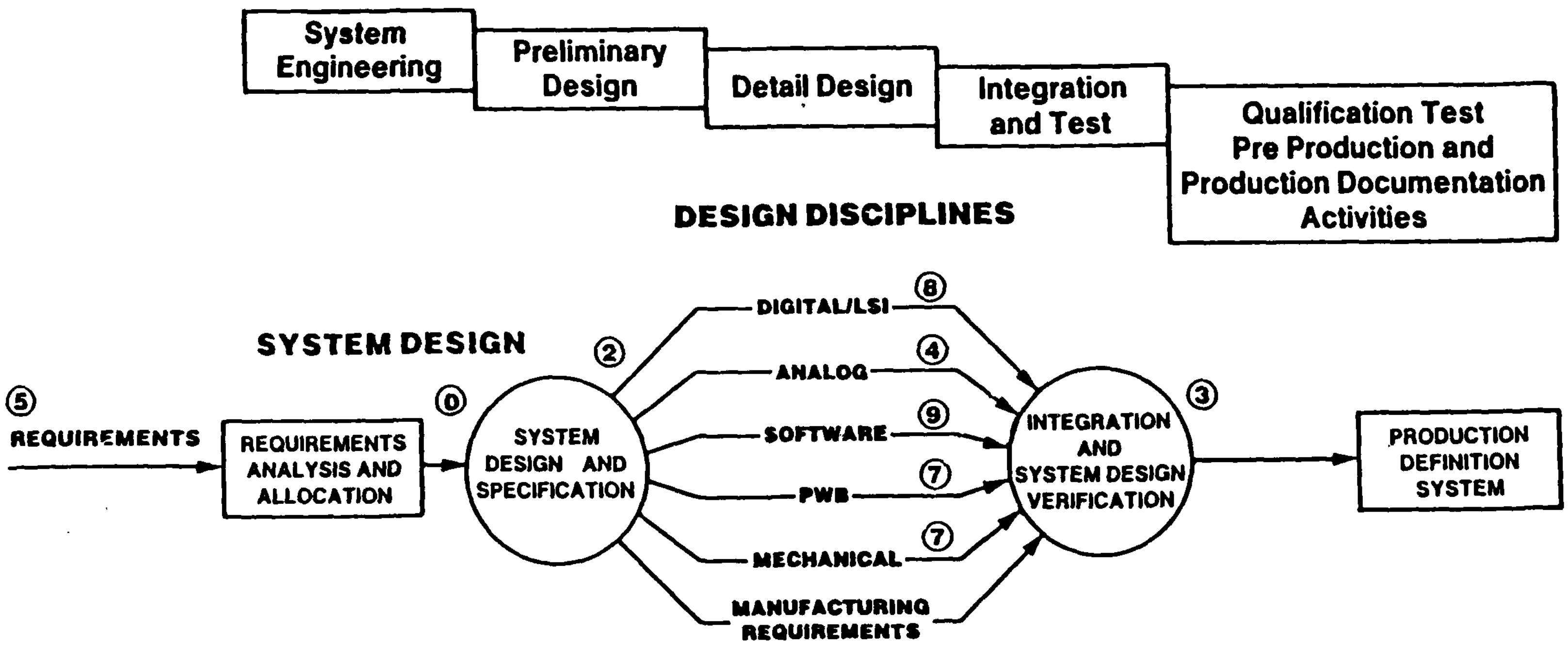
H2274

VIEWGRAPH #10

GEG

KEN PORTER (3-3)
MAY 27, 1987

SYSTEM ENGINEERING DESIGN METHODOLOGY
STATUS OF SYSTEM DESIGN TOOLS



SYSTEM DESIGN TOOL EVALUATION

0 NOTHING ADEQUATE AVAILABLE

5 EXCELLENT TOOL WITH LIMITED INTEGRATION

10 EXCELLENT TOOL AND TOTALLY INTEGRATED WITHOUT CUSTOM DEVELOPMENT

71874-13

VIEWGRAPH #11

H2274

simulate link with any of the tools where you can take English requirements and with some standard high level PDL flow down specification into the different simulators. If one thing is needed, it is a standard language for this process. If the universities really wanted to contribute to system engineering methodology, they could develop a way of accomplishing this function. There is no consistent way of specifying the simulation and design process. Until we get adequate structure and standards in these tools, we're just not going to make much progress in system engineering and lowering the design time and the errors in this sytem design process.

BOOTON: Thank you, Ken. Any short questions? What? [laughter] According to my watch, we've been one hour with two speakers, so with six speakers we'll have minus 30 minutes left for discussion at the end. So I'd like to urge the following speakers to move right along. Barney Reiffen from Lincoln Labs, MIT .... [dropped microphone] We just tested the mike, ah, it still works ....

REIFFEN: I'll try to make up some time. Rather than offer my thoughts on design tools for the future, I think I could be more constructive by suggesting a context for design tools that some others may be proposing. This context will come from a contemporary communication system that we at Lincoln Laboratory are now completing. I'll be much more micro than the previous speakers, I think. The example will serve as a case study which we approach in a very ad hoc way and not the systemized way that everybody's appealing for. But, we did what we did.

Basically, I'm referring to the FLEETSAT EHF package that we have built two of; one of these was launched in December 1986 on TRW's FLEETSAT Satellite No. 7. We at Lincoln provided an EHF package integrated on the FLEETSAT. The first package was launched in December and is working satisfactorily. The second package is scheduled to be launched in a couple of months. FIGURE 1 is a functional block diagram of this system. It is a processing communications package -- is there a pointer that I could use, well, I'll use a pencil ....

BOOTON: I've got one.

REIFFEN: Thank you. Basically we're talking about a communications processor which supports multiple uplinks from a variety of dispersed terminals that are operating at an uplink frequency in the vicinity of 44 GHz. The various uplink signals are demodulated down to data, remodulated on a downlink which is then directed to the various receiving terminals. The downlink operates at around 20 GHz. The uplink modulation and downlink modulation are constructed to be anti-jam and, as such, utilize spread spectrum methods, primarily frequency hopping. The hopping is taken out on the uplink via a dehopper and is reintroduced on the downlink using a different hopping pattern. All of this hardware is configured into roughly a 250 lb package integrated on the FLEETSAT. It operates with two uplink antennas and two downlink antennas. The package has the capability to switch between downlink antennas on a hop-by-hop basis.

I hope this overview gave you a feel for what the system is, and what was entailed in developing it. To give you a scale on it, the process of developing this took about 4 years from the time the project started until, let's say our first package was launched. The job was accomplished by a team of about 60 professional people

and about 200 non-professional people. The cost to realize such a system runs approximately in proportion to the duration of the effort. If you have a delay it just translates into more money, and anything that can be done to speed up the definition, development, and implementation is bound to save money. As I suggested before, the procedure by which such a system gets built, at least the one that we operate at Lincoln Laboratory, is ad hoc. We don't have an overarching approach. We set up a project office; the project engineer runs it, we undertake the system engineering (this is a term that everybody is using, but I'm not sure we all agree on what it means). We try to break all these functions down to assembly units -- boxes.

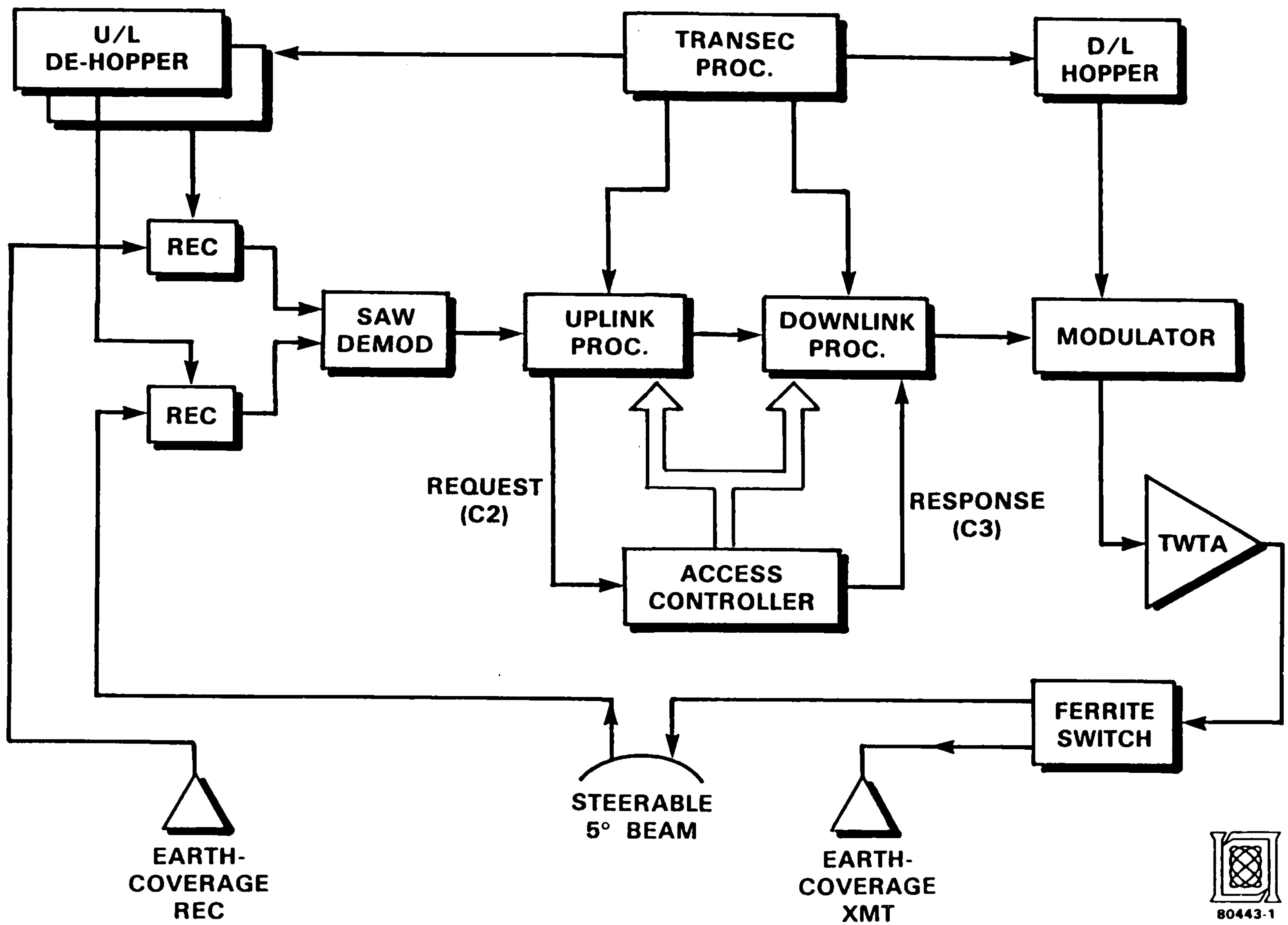
FIGURE 2 presents the signal processing subsystem broken down into boxes, each of which has a unit engineer assigned to it. Each unit engineer has the responsibility to develop his box, build it, test it, integrate it in it's environment. He follows the process from beginning to end, starting with defining the interface specifications between his box and the other boxes to which it connects. This particular unit engineer uses design tools of his choice. We have contemporary computer-aided design for logic design and whatever. There's no enforced design, it's basically implemented in the style of the individual who's doing the job, with the condition, of course, that progress is being accomplished via milestones and design reviews, and it works in the environment of its neighboring boxes.

The total job certainly has a significant design content but is dominated by an awful lot of effort on implementation. Maybe it's 10-20% design, 80% hard work of making the thing work, testing it under the various environmental conditions that the box is supposed to work in, temperature extremes, voltage extremes, after it's been vibrated to make sure it still works, and then fixing all the bugs that appear when you start trying to integrate it with its neighbors. The approach we use is brute force. We don't know of a better one to making sure that a system of this sort works. Particularly in the space environment, it is crucial to test, test, test. It's almost as if there's no substitute for it. There used to be a TV ad about an old craftsman polishing a piece of furniture and an interviewer asks him, "How long do you keep polishing?" and he answers, "Until they take it away from me." [laughter]

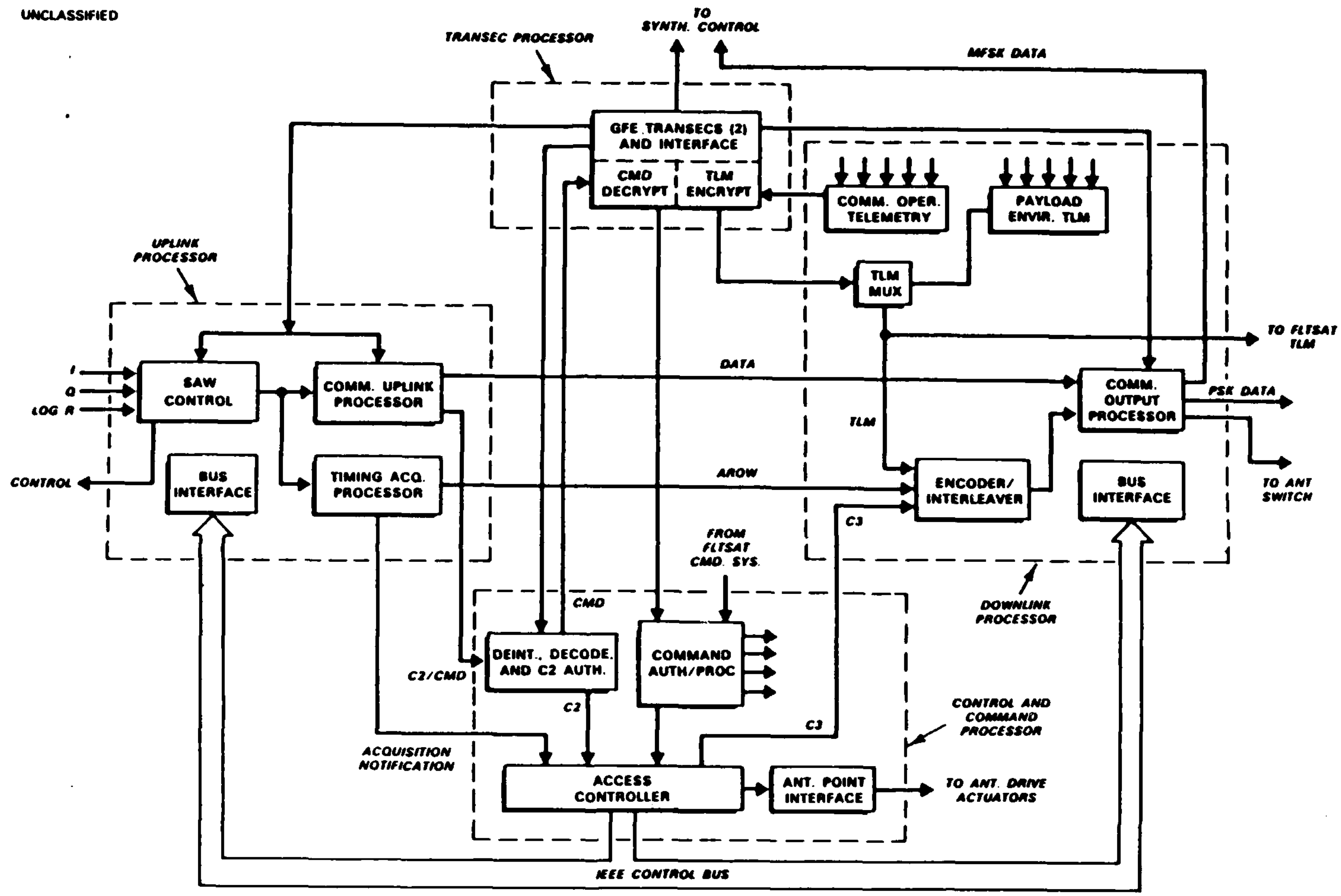
What are the unexpected problems that we encounter in bringing the space hardware into existence? They're dumb, they're stupid. An engineer might have a good design for a box, but it doesn't meet the size or the weight constraints that suddenly appear or perhaps were not apparent to him early on. There were quality control problems in manufacturing the harnesses. Wires came off and defective parts were found on circuit board assemblies. You have to test your parts meticulously to make sure that they meet specs. Another big problem that comes up arises from ambiguities in the subsystem or box specifications. There are interfaces that I referred to before, and these interface specifications are written down in an interface document. There are many such interface documents. Two engineers working on different sides of the interface may interpret the interface differently, and when you try to put their boxes together they don't work. Of course, each box engineer points to the other guy saying, "It's your fault, my box works perfectly." The incompatibility has to be diagnosed, fixed, and then retested. Again, we did all this by

FIGURE 1

FEP FUNCTIONAL BLOCK DIAGRAM



UNCLASSIFIED



FEP SIGNAL PROCESSING SYSTEM

FIGURE 2

142182-I

brute force, and there's no simple way that I know of to avoid it. I hope there are some simple ways that you can come up with to do this sort of thing.

I'll conclude by saying that this is not the kind of a system that gets built in large production quantities. Only a few of these will ever be built. It's not a system that is going to serve a massive community and major reconfigurations are going to be done over its lifetime. If that were the case, perhaps it could justify a large software support system for figuring out the optimum reconfiguration. The FEP system I have described occupied a large number of our best hardware engineers and communication system engineers. I offer it as an example, as a potential system to test design tools for the future to see whether we could have done this sooner, better, quicker, or whatever if appropriate tools were available. Thank you.

REY: How many system engineers did you use out of that 40 professionals for this project -- 60 professionals?

REIFFEN: A very small number. It's very hard to distribute the system engineering task, and actually at the core of it there was probably no more than half a dozen, ever; and even in the early stages of it, fewer than that.

BOOTON: Thank you, Barney. Barney warned me he would say something different this time, and I appreciate it. One announcement -- we're not going to take a formal break, but there are coffee, rolls, and watermelon slices in the back, so please eat your share and do it informally. We're doing fine on time now, but ... yes?

AMOROSO: It is an axiom of experience that when you set a test procedure, you're mostly guessing what might go wrong, what might probably go wrong, what might probably be at fault and, you're checking against those specific guesses. Is there something more than that at the base of setting a test procedure?

REIFFEN: I think not. I think that what one wants to do in this context is test everything. I didn't go into detail on this point, but this is a system with multiple uplink and downlink modes which can function in many combinations. We would be testing from now to doomsday if we tested everything. So we try to search the space of possibilities in some sort of reasonable way. When we anticipated a problem, we worked on that well enough so that that's not where the problem was. [laughter]

BOOTON: Okay, let's move along. Raul Rey is going to talk next about some of the work at TRW in this area. I should explain that Raul played a very important part in a workshop I had last week in Dayton, and that Raul gets very enthused about his subject, so ... I may have to drag him off the stage at the end to hold our schedule ....

REY: I'm not going to talk that much about BOSS, although I do mention it a couple of times in the talk; I'll let Dr. Shanmugan give you the details. You see, Barney really led into my talk very well, we do the same kind of system engineering and build a similar kind of hardware at TRW. We build primarily space communications payloads and systems, groundstations; they're large systems, they're very complex. If you look at what happened with our development of TDRSS (Tracking Data Relay Satellite System), which at the time was considered to be a very complex payload, and then compare it to our MILSTAR payload, the complexity of MILSTAR is around five times greater. You can

see it in the number of engineers that were used to do that job, and also the number of system engineers -- I think there were somewhere around 20-25 payload system engineers on TDRSS and with the more modern analysis and simulation tools that we had for MILSTAR we still had around 75 system engineers on MILSTAR.

What I'm going to talk about are the engineering tools that we have today and how we're integrating them (see CHART #1). I won't talk about the system engineering process, because I think Barney did a real good job and the previous speakers have described that process very well. I want to talk about our tools, analytical tools and how we're integrating them. I want to talk about the system simulation tools and how we're going to integrate those with the hardware designer's tools, talk about our hardware database concept that we're moving towards, and then some applications.

Now one of the things that I see that's happening is that originally the CAD-type tools were being developed by and for hardware people who were interested in the very low level details of developing hardware. There was not a lot of thought towards the system engineering aspects of those tools, and what I'm seeing, the thing that's beginning to happen now, is that the whole process is being integrated. Even the hardware people are looking at tools like BOSS to see how they can use a simulator to functionally design their units and then integrate that with all of their CAD tools.

What we're doing at TRW is we're defining the tools that we're going to use -- the CAD tools that we're going to use in the future -- we're writing specifications on them, and then we're going to buy a CAD system or systems (plural) that we'll use to design our future systems. Our tools are going to be integrated and, in fact, are being integrated now. Currently, we have a number of simulators, our network, our link simulators, orbital-digital simulators, and one by one these is being integrated into one simulation system. We have a lot of analyses programs that we use to predict things like IM performance, spur performance, bit error rate, all these different parameters.

At TRW we do our job differently on different projects, so I can say at least one project already has analysis programs that are integrated into their technical performance monitor software so that if you run, let's say, a G/T analysis program and get a result, it'll go into the technical performance monitor program and then calculate automatically the performance of the whole system. Now, in some cases, that's done manually in some programs where someone will make a calculation, take it from one computer to another computer, manually put the data in, run the TPM (Technical Performance Monitor) program. The process is very cumbersome. (CHART #2) Data is being handled and interfaced with paper when electronics is available to do the transfer via communications lines.

Our CAD tools are being integrated with unit and system simulation programs. One of the rules we have is that the CAD software that we're buying have to interface both with the system and unit simulators, and we're defining those interfaces to get that process started. We're developing a hardware performance data and model database. The reason I say both hardware performance data and model is that what we need to have is a database that's accessible to system engineers and unit engineers, that contains not only performance data but contains the models for each of the pieces of hardware that we're developing. It has to

be accessible to not just one system engineer but to each system engineer responsible for a particular payload. One of the things we're doing is we've got a road map for system simulation. We've got one project where we're working with the University of Kansas to study the techniques of system simulation by developing and applying BOSS. We've gone through a number of versions; the next step here is to develop a network simulation program that has a BOSS-like graphical driven simulator. We have some optical simulator models that we're integrating into our overall simulation capability. But the final result is to have an integrated system for simulation.

An example of what we're trying to do is in the area network simulation, where the ability to transmit data through various links is a function of the bit error rate on that link. If you have a low bit error rate, you must retransmit. So what we're trying to do is have direct interfaces in our communications link simulators to generate the BER data that is passed up to the network simulator. The network simulator can ask for data of the lower level simulator.

CAD tools are available to unit designers, they can generate layouts and emulate hardware functions and generate performance data. Now two of those functions, as system engineers, we can use. If they've got software that can emulate hardware functions, we would like, as system engineers, to get those models and put them into our system simulators. They should be compatible models. Also we would like to be able to have access to the performance data that hardware engineers generate as they're designing their hardware. They're predicting the performance of their hardware; if we can feed that into our system simulations, we can predict or better predict the performance of our systems. (CHARTS #3 and #4)

BOSS, we believe, is an excellent tool for the functional simulation of units. It actually, although it was designed to be a system simulator, is really -- sounds like a sales pitch, but -- an excellent simulator for units. It does all the things that you need to do when you specify a unit. An analog type unit has to have transfer functions, input signal versus output signal, looking for spurs, gain ripple, which is how we spec a unit. BOSS does a really good job at simulating those kinds of functions, and we expect now that hardware engineers are going to use BOSS to make the first cut designs of their boxes before they even get down to the CAD level detail design of the units. The unit models and the data that is generated by hardware people to test their designs are the very same models that the system engineer can use in his higher level system engineering system simulation. So what we have here is a nice interface, I believe, using BOSS as the core -- I'm getting a little ahead of myself here because that's the decision we've made -- is that BOSS will be the core software for our new CAD system, the system we're going to specify.

(CHART #5) We at TRW have spent just about a year now looking at our computer-aided engineering process. The goal is to define and specify one unified system that we can use throughout project phases, and we've formed a number of committees which have looked at the process for different disciplines of developing electronic equipment, RF, low speed digital, high speed digital and antenna disciplines. (CHART #6) Those committees have decided, at least three of the committees, that the antenna people have not come in with their decisions. The other three committees have evaluated the process and

decided that BOSS will be the core for our computer-aided design process, and that all future procurements of CAD tools will have to have interfaces to BOSS that will take data from, and pass data back to the BOSS simulator.

(CHARTS #7 and #8) One of the keys to the whole concept, I believe, of integrated tools is the hardware database. The key here is that we need to have good, accurate data and models that we can use to predict the performance of our systems. Now the interface between system engineers and hardware engineers is cumbersome to get data and give data, it's a very cumbersome process. Whereas if we could get an electronic database where we could go in and by using the right kind of cataloging technique call out for a model for a piece of hardware, we can get the real data, the real representation of that hardware as it was either designed and the data presented as CAD data or test data that comes out of a unit test. It is really important that we have that kind of capability. Something like this isn't only used in the very beginning part of a program, it's used throughout the program, all the way from where we do our evaluation of the design, as a unit engineer does his CAD design and evaluates his design. (CHART #9) A system engineer can take that data, put it into a system simulation, as a unit performance changes, as the design changes, we can look at the system performance changes. (CHART #10) There are times when we would like to have our systems characterized well enough that if we have a problem on orbit, we can go to that hardware database, pull out the representation of that system, run simulations to test whatever we believe the failure to be on orbit, we can simulate it on the ground if we have an adequate characterization of the spacecraft payload.

One of the things that we came up against right away was a typical engineering question, that any analytical engineer would ask: "Why do we need this?" Well, there are the obvious reasons of cutting down engineering time, more productivity; but there's also a really overriding, I think, important reason, credibility with our customer. That if we have good data taken from hardware units, good simulation models, we gain much more credibility with our customers to show them that if there's a unit out of spec and we're claiming that the spec on the overall system is being achieved, he can rely on that data. I think that's a primary need, because it's important to us to maintain our reputation with our customers.

(CHART #9) This just shows, in kind of a flow-diagram fashion, what the concept is with a hardware database. Probably the key thing here that's different than what you've seen before is the use of BOSS as a unit simulator. It provides the key interfaces between the CAD design tool and the unit that's under development. Data from the CAD or the evaluation goes to this hardware database, unit data from tests that are run go to the database, the models are in here, and it's easy for the system engineer to call up the appropriate models for a unit. The concept here is that every unit as we develop it is catalogued. We have these units catalogued by what we call equipment specs. We can go to our model library, we can call up that unit, that component by its spec, and there should be a model for it and data that we can use in our simulations. It may sound like there's not a lot to be gained there, but you'd be surprised how many times in the development of a system that we have to take a unit out of a payload and put it on a bench to have it tested and replace it with another

unit. When we do that we have to predict the performance of that channel again, and it takes a lot of work if we don't have adequate or the right kind of data that characterizes the new unit. So I thought I'd talk about some of the applications of this integrated approach, and particularly the hardware database, because there are some real benefits to our customers.

One of the things we would like to do is have a rapid prototyping capability. (CHART #10) At TRW we have years of history of building spacecraft payloads and communications systems. We have a lot of data for all the hardware we've built; unfortunately, that data is in databooks all over the company and very hard to get hold of. When we propose a new system to a customer what we would like to do is to put together a simulation of that system, do a rapid prototyping of it by putting a system together, show the customer what existing hardware we would use and what its performance is, then put in only that new hardware that is needed to solve his problem, and then demonstrate to him that we've got a working system and it'll meet his requirements. This has been done by software people at TRW. In the Defense Systems Group they have a facility for rapid prototyping. They can put a system together, bring in a customer, show him the performance of that system, not at our level, the communications level, but for building computer systems. It's been very effective.

(CHART #12) Let's see, I won't go through the whole process here ... or Dr. Booton will throw me off the stage. [laughter] I thought what I'd do here is show our concept of an automated system for performing system engineering. And although it looks very repetitive, each one of these processes is actually different. When we start off we have a database of just models, predicted performance. The system engineer has to take that data and predict that the performance of the customer's requirements are going to be met. At that time he starts to negotiate with his counterpart hardware engineers. There are questions of interfaces, whether or not the unit can meet the requirements being imposed on it; and each time this negotiation takes place, where a hardware engineer, let's say, gives a tenth of a dB, we have to redo our predictions of performance. This is a rather involved process, and it happens all the way through the design phase and we do a lot of negotiating with our hardware engineers and our customer, and manufacturing is a stage that takes a lot of effort; a unit will come out of testing and it'll have some discrepancy -- it doesn't meet some spec like the gain ripple, phase distortion -- specs that w can't just write off. We have to go to our customer and prove to him that first of all the box is not broken, that's the key thing. But the second thing we have is that if we accept that unit and put it in a system, we have to convince the customer that we're still going to have adequate margin in system performance. So we go through this process and all through this process we use simulation quite a bit. Now in a sense this isn't a new concept because we did it on other programs, on TDRSS we use LINK to simulate the payload. As data came in from units we replaced theoretical data with actual unit data, but we did it manually. We read the data out of data books, and put it in the simulation. It always seemed to be a little silly to me because the data was taken on computers, it was on tapes, and I didn't know why we had to continually do this manual transfer. But we've done the process before and the big difference here is to

TRW

INTEGRATION OF ENGINEERING DESIGN TOOLS

R. D. Rey

27 May 87

TRW

ENGINEERING DESIGN TOOLS WILL BE INTEGRATED

- Effort is being expended toward unifying system simulation and analysis software

> Network	> Technical Performance Monitor
> Link	> Link Performance Budgets
> Orbital	> IM, Spur, G/T etc.
> Digital	

CHART 1

TRW

ENGINEERING DESIGN TOOLS WILL BE INTEGRATED (Cont.)

- Cad Tools will be Integrated with Unit and System Simulation Programs
  - > CAD SW must be compatible with and interface with System and Unit Simulators
  - > Developing a Hardware Performance Data and Model Data Base is necessary.

CHART 2

TRW

SIMULATION in SYSTEM ENGINEERING PROCESS

CAD Tools are Available to Unit Designers for Detail Design

- Generate Layouts
- Emulate Hardware Functions
- Generate Performance Data

BOSS is an *excellent* Tool for functional Simulation of Units

- Simulates Transfer Function
  - > Signals vs Signals out
  - > Gain Flatness/Ripple/Phase Linearity
  - > Spurs/IMs

CHART 3

TRW

Simulation in System Engineering Process (Cont.)

- Simulates Algorithms and Digital Processing
  > Digital Timing and Performance
  > Algorithmic, Functions and Processes

Unit Models and Data can be used with BOSS to perform System Performance Evaluations

CHART 4

TRW

COMPUTER AIDED ENGINEERING PROCESS AT TRW

- Committees Formed to Study, Define Process and recommend Procurement of Hardware and Software

  > RF

  > Low Speed Digital

  > High Speed Digital

  > Antenna

CHART 5

COMPUTER AIDED ENGINEERING PROCESS AT TRW (Cont.)

- BOSS has been Selected to be the Core Software of CAEP
- All new CAD Hardware and Software must Interface with BOSS
- All Technical Performance Monitoring (TPM) software must interface with BOSS

CHART 6

TRW

HARDWARE DATA BASE

- Hardware Data supports Systems Performance evaluations
- Performed through All Project Phases
  - > CAD Evaluation
  - > BOSS Unit Simulation
  - > Bread Board Data
  - > Engineering Model Data
  - > Quality/Acceptance Data
  - > Integration and Test Data
  - > Operational Data

CHART 7

TRW

HARDWARE DATA BASE (Cont.)

- All the above data reside in Hardware Data base and is accessible to BOSS for System level Evaluations
- Benefit System Performance Evaluations have increased credibility

CHART 8

System Simulation In Design Process

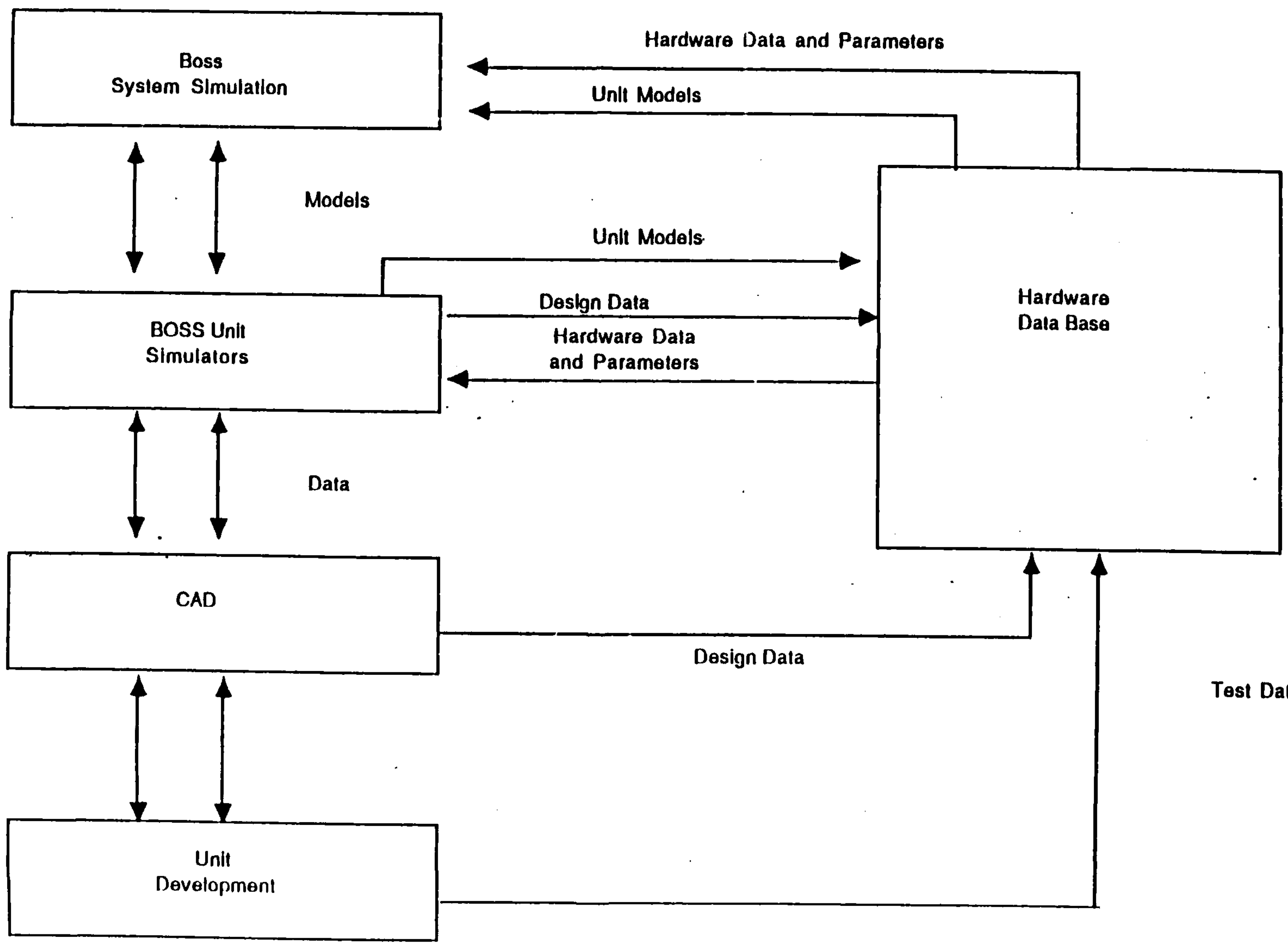


CHART 9

TRW

APPLICATIONS

- Rapid Proto-typing
  > Configure and demonstrate new designs for customers
  > Use existing hardware designs with new designs
  > Modify designs

- Preliminary Design Evaluation
  > Determine system inpact of specification allocation
  > Evaluate system performance impact of hardware requirement changes

CHART 10

TRW

APPLICATIONS (Cont.)

- *Out of Spec Evaluations*
  - > Breadboard
  - > I & T
  - > Manufacturing
- Sell off System Performance to Customer
- Evaluate *on-orbit performance or anomalies*

CHART 11

System Performance and Hardware Simulation Data Base

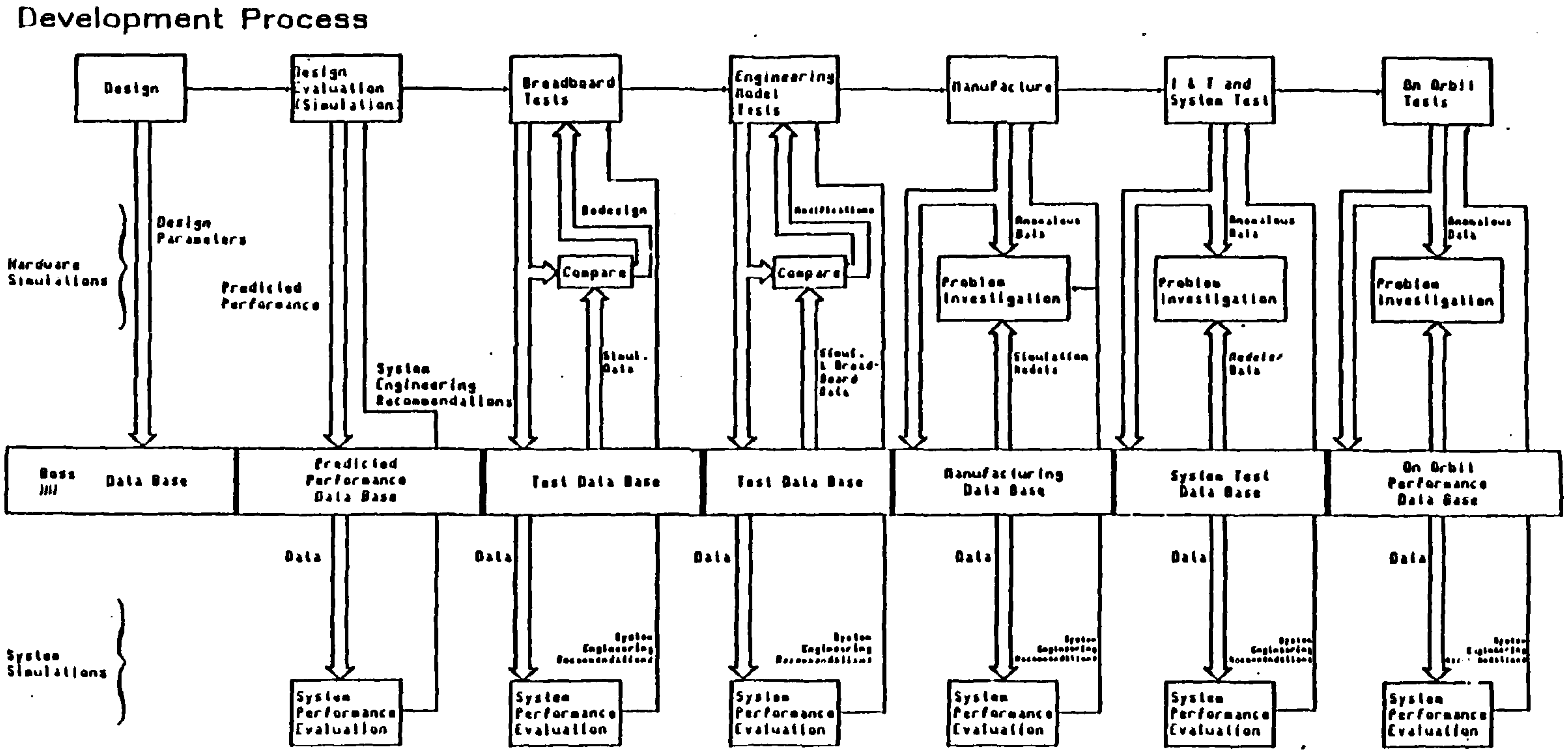
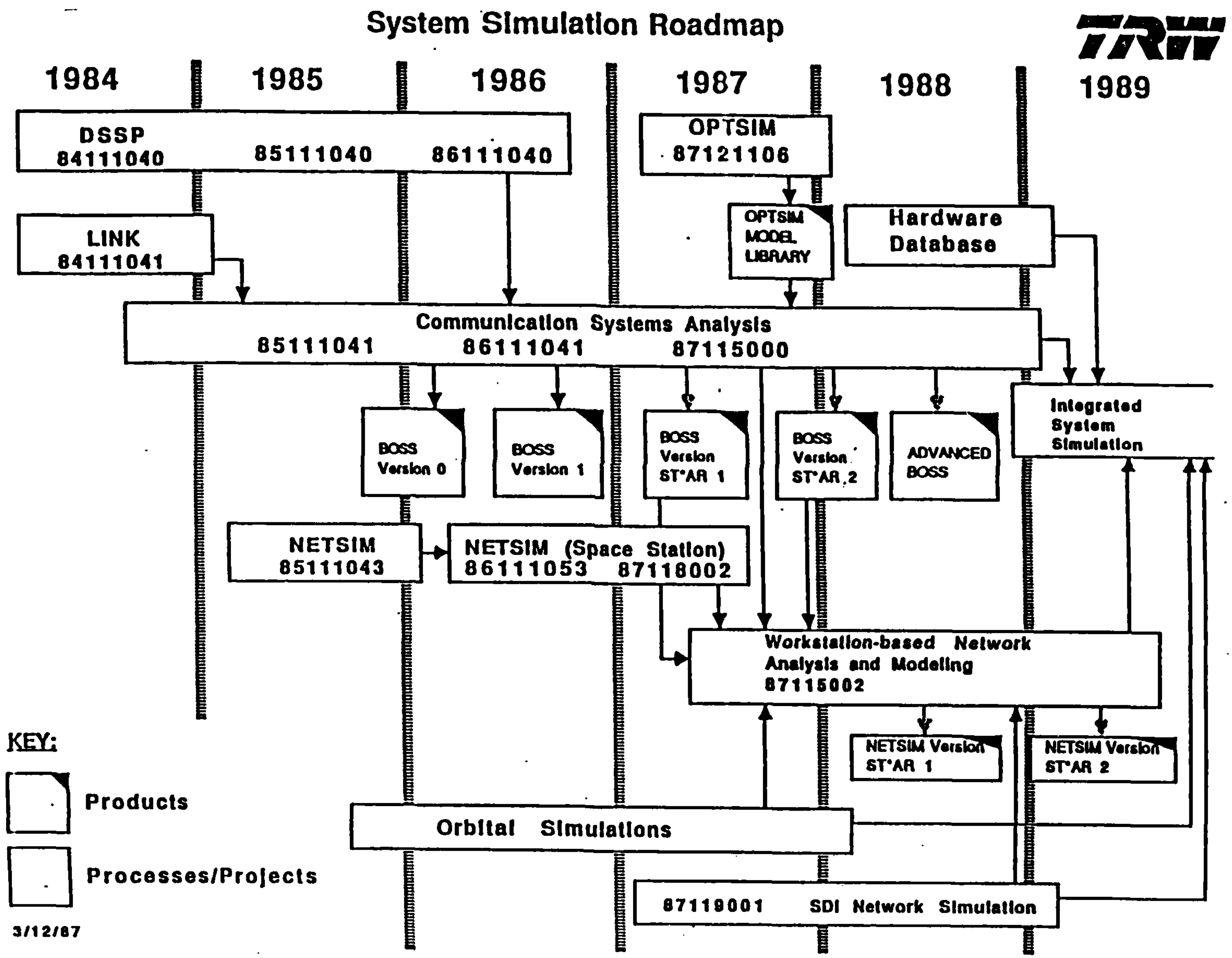


CHART 12

R.D. Rey
9/17/86

CHART 13



System Simulation Roadmap
TRW
1984
1985
1986
1987
1988
1989
DSSP 84111040
85111040
86111040
OPTSIM 87121106
LINK 84111041
OPTSIM MODEL LIBRARY
Hardware Database
Communication Systems Analysis
85111041
86111041
87115000
BOSS Version 0
BOSS Version 1
BOSS Version ST*AR 1
BOSS Version ST*AR 2
ADVANCED BOSS
Integrated System Simulation
NETSIM 85111043
NETSIM (Space Station) 86111053 87118002
Workstation-based Network Analysis and Modeling 87115002
NETSIM Version ST*AR 1
NETSIM Version ST*AR 2
Orbital Simulations
87119001 SDI Network Simulation
KEY:
Products
Processes/Projects
3/12/87

try and get some electronic interfaces to get us the data we need and increase our productivity. Thank you. Any questions?

HARRIS: This may be more of a comment, but, one of the things that you said that really rung a chord with me was the need to do rapid prototyping. That's something that's becoming apparent in our industry, that we're going to have to be able to prototype new services, new products and new capabilities a lot more than in the past. We find Japan, for example, does a lot more prototyping of advance systems than we do in this country. One approach that we're taking, I'd just like to mention it because it involves something we're doing with a university, we've found in looking at network prototypes that building the prototypes are too expensive usually, and simulations are good but they have some problems, and we've gone to network emulation a bit. There's a project at Berkeley that's being headed by Prof. Varaiya and Prof. Warren called the PNPS or Programmable Network Prototyping System, which Pacific Bell is collaborating with. It is software-based, and will allow the emulation of network nodes, network transmission and network hosts, all defined through a software interface. That is an approach that -- I don't know about that particular project, but -- that's an approach that we feel is important in terms of tools, are tools to allow emulation and quick prototyping in the future.

REY: Well, in the space payload business, the way government budgets are going, we're not going to be allowed to build total brand new payloads. It's going to be a very competitive business in the future and we're going to have to use existing designs wherever we can. At TRW our competitive advantage might be that we've already built a lot of space qualified hardware and we want to use that in our future designs.

SCHOLTZ: Raul, could you comment on the size of the database that you expect and how you plan to manage this in the future ... it seems to me that this could become quite a monster after several years of operation.

REY: We've thought about it but we haven't sized it; however, the only way to keep something under control is to use the present functional structure that we have. Each laboratory builds a certain set of hardware, so each laboratory has to have a database for their hardware. If we know where the hardware was built we can access their computer and data base. What we're doing now is we have a number of VAX 780's that are in the digital design center, we'll have something in the system engineering laboratory. The only way to keep this under control, I think, is to let each functional element, each product area, control their own database and then we just have to know where to access the data. If it's RF equipment, you go to the Com Lab database; if it's digital, you go to the Digital Lab database.

REIFFEN: I would offer a comment, and perhaps a concern on the theme of database and sizes. With an integrated system, along the lines that you propose, there is a concern -- I guess I just gave it the name the Wheat or Chaff problem -- the unit engineer is interested in aspects of his box behavior which is very detailed. I would offer aspects of its reliability, of its marginal design, and so forth, which are not of concern to the system engineer. In putting together an integrated system, it seems to me, one must be aware of the potential of confusing the system engineer with chaff. The amount of information that

the system engineer wants to know about the functioning of the unit is generally very much less than the amount of information that the unit engineer wants to know.

REY: Yes, I agree. I think that's what makes the concept of BOSS as a unit simulator at the functional level, rather than at the detailed component level, a nice interface. The unit designer will design his box first as a set of functions rather than getting down to the detail, and that's where he'll interface with the system engineer. I don't think the system engineer needs to know the CAD portion of design. He would like to have data to characterize the unit, for instance a channel. He would like to have the frequency response for that unit, and if in designing the box the hardware engineer has produced that data the system engineer would like to have access to it.

BOOTON: Thank you, Raul. And I especially thank you for not forcing me to drag you off the stage .... [laughter] Very good. I can't help it, I had a comment though about rapid prototyping. We've watched our software colleagues, as Raul said, working on rapid prototyping as a tool for rapid simulation of a major software package, and one of the major benefits has been that they can bring a customer in and show him how the system is going to work, and quite frequently they find out that he doesn't really like it. He gives them specs and they go off and rapidly put together a simulation system, and then find out what the customer likes of it. In the past they would start with the spec, go off, spend several years and a few million dollars, and produce a software package, and discover that it wasn't what the customer really had in mind. It was what he put in the specs, but it wasn't what he had in mind. We've done that on hardware programs; we developed a microwave radio for the Air Force to use in Europe and Korea, and it wasn't used by the Air Force for reasons only DoD understands. The DCA wrote the specs and it was an Army contract, and we spent about three years and quite a bit of money developing the system, and got the final test results, and sat down with the Air Force and the Army, and the Air Force says, "We can't use this ...," and they gave us the reasons. If we could have had some kind of rapid prototyping in the simulation sense to show the customers in the beginning how the system was going to operate ... that may be way off in the future to develop hardware rapid prototyping in a simulation sense, in the same way that software people have, but I think it's a worthy goal. Enough of that. The fifth speaker is Jim Spilker, President of Stanford Communications.

SPILKER: Thank you. As the last speaker I have the advantage that all of you are still here, the hard core. The problems that Stanford Telecommunications have fall into the same category as the two previous speakers from TRW and Lincoln Lab -- albeit on a slightly smaller scale. We have our organization operating almost entirely on fully projectized teams with the system engineers doing the overall system design and then coming back heavily in the test and integration phase. The hardware and software people are on the same team with a considerable amount of our work being done on ASICS and microprocessors.

When Dick asked me to say a few words about design tools for the future I was debating with myself as to whether to talk about some of the design tools for system engineering, which I think are moving along fairly rapidly. We have now the ability to take an entire telecommunication line, including a very complex transmission channel with disturbed ionosphere and pro-

pagation disturbances of that nature, digital signal processing, on both the transmit and receive side, and fully to simulate the entire network with the supercomputer. I think that process, in other words the entire simulation process, is going to be more and more powerful now that we can have 1/4 the speed of a Cray at some fairly reasonable price. A second set of design tools is used in the development of microprocessor software and our large Mainframe software where we have very powerful workstation tools linked with one another and supporting teams of software engineers.

The area about which I finally decided to talk, however, is in a third category, VLSI design tools. I would like to look at some of the tools we have now and some that we will have in the future, and present basically one person's view, my own, of where we're going in digital design.

I see the digital semiconductor world of the future as really having only three different families of components (see FIGURE 1). The first being families of microprocessors, the second being memory elements, and the third being the Applications Specific Integrated Circuits (ASIC) and this latter, I think, is going to be one of the dominant types. In fact, I would predict that of these three different classes of integrated circuits, that ten years from now ASICs will be the dominant type. I'll probably get a lot of argument about that but anyway that's my prediction.

I would further predict that the A to D and the D to A converters within those systems will be integrated on board the ASICs. We can do that right now as a matter of fact. So you can have a system where in the real world the processing done in a receiving system starts out with an A to D converter operating at some reasonably high intermediate frequency of the receiver, never down converting down to baseband at all. That A to D converter would operate at what some people consider to be a radio frequency, especially if you happen to be in the HF radio business. So you basically are converting to digital very early in the game. We will have ASICs that can operate at processing speed up to 1 to 2 gigawords per second. We can do something close to that right now, but with a limited number of gates.

Now in looking at VLSI as a design issue and at the history of what's been happening over the last ten years, it seems to me that we're in a very exciting area of transition. If I remember back in the 1960 era, a few years after I received my Ph.D. from Stanford, the system engineers and analysts were many steps removed from anyone actually building hardware. That isolation I think has been progressively disappearing.

Along with that disappearance of the isolation, we see a rapid increase in the complexity and capability of ICs. In the 1960 era when ICs first came out in the planar type forms you would be lucky to get maybe four to a half a dozen transistors per integrated circuit chip (see FIGURE 2). Now in the last 1980s we're at approximately a million transistors per IC chip, and depending upon what you project for the future with the standard single layer types of integrated circuits, we can see that rising above a million, perhaps up to 16 million. There have been some projections of even higher numbers of multilayer integrated circuits or three-dimensional ICs which are now starting to creep up in capability to the same number of transistors per IC chip as we have with the more conventional type. Thus at least some people are projecting that complexity will rise on through the

level of a million gates per IC chip. Certainly we will have integrated circuits of really tremendous complexity, particularly if you talk about three-dimensional ICs where we're stacking multiple active integrated circuit layers, separating them by various types of dielectric insulating materials.

Now if you couple that increase in gates per IC along with the increases in speed, you have some tremendous capabilities, but you also have some enormous design problems. How do we manage that complexity? How do we manage, if you will, the coming crises of complexity of the devices when we're growing to more than a million devices on a single chip? If you look around the countryside, you'll see that in fact there are a number of different successful systems that have well over a million components; for example, a hundred story skyscraper is certainly going to have millions of components, wide body jets, the same thing, and obviously the nationwide telephone communication network (see FIGURE 3). One of the key things that you'll recognize in these systems is that there are very successful partitionings of a physical nature in them. In the VLSI chip I think we're going to have to do the same partitioning perhaps not so much physically as you have in a skyscraper, but rather a logical partitioning. We will also see a very rigorous structuring that we have found necessary in the complexity crises for software, which we faced some years ago.

I think there are things to learn from the lessons of software complexity encountered when we reached complexities of several tens of thousands of lines of code. When we got beyond the point where one super wizard could write all the code and keep track of it, you began to find that you had to have much greater structure in that software, and that you required teams of software people. Rather than relying on one single wizard you'd like to have several wizards working in parallel. You also found that those teams really have to be reasonably small, otherwise you face the problem that you have so many people working on the task that they spend all of their time communicating with one another and never get a damn thing done. So keeping the levels of communication between the team members manageable is certainly a key issue. At least some people have suggested that the use of electronic mail between team members is a step in that direction because it is time efficient. Thus I believe we should keep the teams small to minimize the communication requirement and aid them with design tools that can enable them to stay small.

Now one of the problems that isn't talked about or taught very much and for which there really isn't a good design tool available is the task of successful partitioning of the functions on a VLSI chip so that those functions, those subfunctions or modules, those macro cells if you will, can be compiled automatically or semi-automatically on a silicon compiler or some higher order language. Now at UC-Berkeley, C. H. Sequin has really been a proponent of that; as a matter of fact, proposing not only the use of design tools but greater course structuring within the universities, teaching how to manage complexity. I argue that this training really is an issue that has to be dealt with in addition to all the other design tools that one might have available.

There are some comparisons that one has in looking at VLSI and software design. First of all a software programmer and VLSI designers have some things in common in the way of productivity. For exam-

ple a software coder can code in the ballpark of ten lines of code per day (see FIGURE 4). I think we all know that there are some software wizards that can maybe do double that many lines of code, or perhaps even more. However, in either case the lines of code produced have been found to be surprisingly independent of the type of code that you're using. For example, you can do five or ten lines of higher order language code, or you can do ten lines of assembly code in the same time. The complex algebraic higher order language statement, of course, has a great deal more in it than an assembly load instruction. So the efficiency of those two types of operations, one using HAL and the other using assembly languages, is markedly different.

Now with ASIC VLSI design, I think you have much the same situation -- you can argue about the number, but -- perhaps ten "items" per day is what a standard ASIC designer can do. The efficiency of that design again varies largely depending upon whether his "items" happen to be ten processors per day or ten macro cells, if you will, or whether it be simply ten transistors per day. So the statement here, just as in software, is that the key is in partitioning the system into macro cells and then proceeding so that the designer operates with the larger cells, and then automates the lower level design of these macro cells. I think there is at least one difference between the complexity of statements with which one has to deal in software and hardware and that is the hardware ASIC design is at least a two-, perhaps a three-dimensional type problem whereas the software is perhaps at least in some sense has only a one-dimensional problem.

Now let us look at the cost of the ASIC chip (see FIGURE 5). There are a number of different ways that one can build an ASIC chip, programmable logic arrays -- I didn't really list them here, but those certainly are one form; gate arrays where you basically have a cell layout and you have one or more metallic interconnection layers overlaid; the standard cell which is an intermediate step to a fully custom handcraft design. In the full custom design everything on the chip is pretty well free for the designer to use. If you look at the cost of these alternates the overall cost per chip is basically the development cost divided by the number of units that you're going to produce in this system, plus the production cost of the cell which is directly proportional to the die area, the die size if you will. So that if you sum those two then you end up with the average cost of your chip. The die size varies significantly for the same logical elements depending upon what you h ·e in the way of design technology whether it be gate array, standard cell, or full-up custom. In one reference design that has about 6,000 gates and 1.2 micron technology, we end up with a die size reference level of 1 for the gate array and .4 for the full-up custom, and a little bit more efficiency for the standard cells than the gate array. The development time, however, is the quantity that varies most dramatically with the gate array at a reference level, x, which typically might be on the order of 3 months; 1.5 times that for the standard cell, and then the full-up custom could be as much as 6 times that. I argue that as the complexity of the chips becomes even larger than the relatively simple 6,000 gate chip that we described here, that this full custom number could be even larger compared to that for the standard cell and gate array.

Now the effectiveness of design using macro cell higher order design language tools or silicon compilers, as you might choose to call them, basically vary all over the map at this point in time. And if you talk to various people about silicon compilers you'll find that there's a very mixed review there at this present time. I think that the performance will improve dramatically as time goes on but perhaps not to the same level that it has with the software arena. You can now get some very efficient software compilers that do not perform too far below someone writing directly in assembly language. But with a silicon compiler -- I should say a gallium arsenide compiler as well -- you end up with die size figures which are strongly related to the regularity and structure of the chip that you're designing. For a very simple structure like a ROM you can typically get 30,000 transistors per square millimeter, for a RAM, almost as many -- about 10,000 (see FIGURE 6). However, as you go to random logic at this point in time you achieve a very low efficiency relative to the more heavily structured units. For regular logic, perhaps an arithmetic logic unit with some degree of structure, you can do better than random logic but still not really very good. So a lot of progress is yet to be made, and we see quite a number of different companies trying to work in that arena.

As a communications systems engineer I always looked towards the time when you could sit down and write out your equations (see FIGURE 7) and put those equations and algorithms into some sort of simulator to ascertain that those in fact are the right algorithms, that they have the right degrees of quantization and so on, the right clock rates, the processing that you want to do is being done, that some tricks and the algorithms that can allow you to speed up the process are in fact all performing to your satisfaction. Once you've reached that point you would hope that you can now take that algorithm specification and feed it into this array of VLSI design tools that would include such things as supercomputers and perhaps supercomputer-aided workstations, superaccelerators, if you will, to aid in the simulation of the VLSI process.

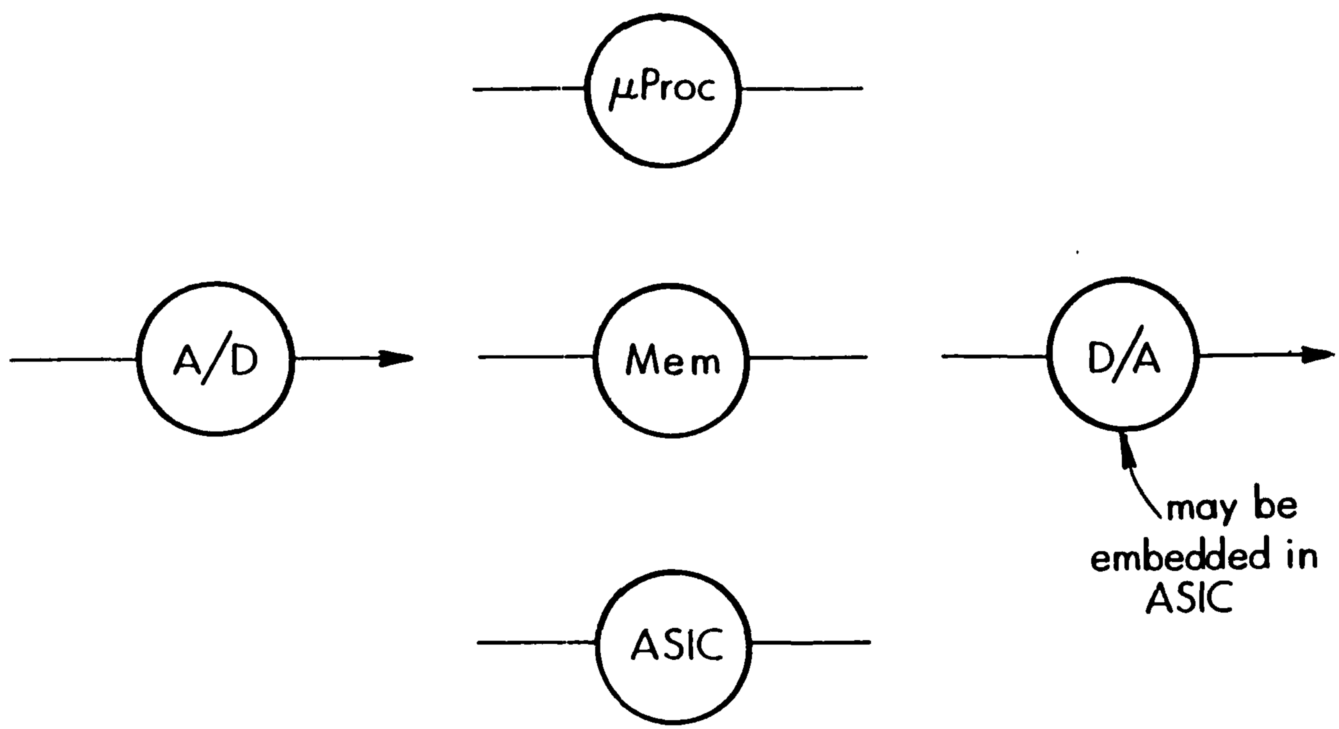
Expert systems software hopefully will take care of some of the heuristic elements in the design of a true expert VLSI design engineer. I don't think anybody expects that the expert system software is going to be as good as the expert, but perhaps if it can be 80% as good that would still be a tremendous accomplishment because you don't have that many experts floating around the world at this point in time. Anyway, if you can simply do 80% as well as the true expert and multiply that by using many hardware processors, you're certainly much better off. Next the silicon compilers operate on the partitioned circuit, and finally mini-supercomputer simulation and self-testing being the two other crucial elements of the future design process. Now in this particular chart I didn't really list the elements that have to be done in the foundry because that's a whole new set of software tools that are more related to the physics of the operation and the process control and so on.

In any event, what I would hope to see is the design tools that we're proceeding with now enhanced to the point where we can in fact have that very close association between the communication system engineer and the foundry. If you look at some of the things that I think we need to do in order to get there, first of all, we now have a great many different CAD-CAE

VLSI DESIGN TOOLS FOR NOW & THE FUTURE

Jim Spilker, jr.

One View of Future Digital/Analog Design



3 Types of Digital ICs

FIGURE 1

Growth Trends in IC Complexity ≈1M

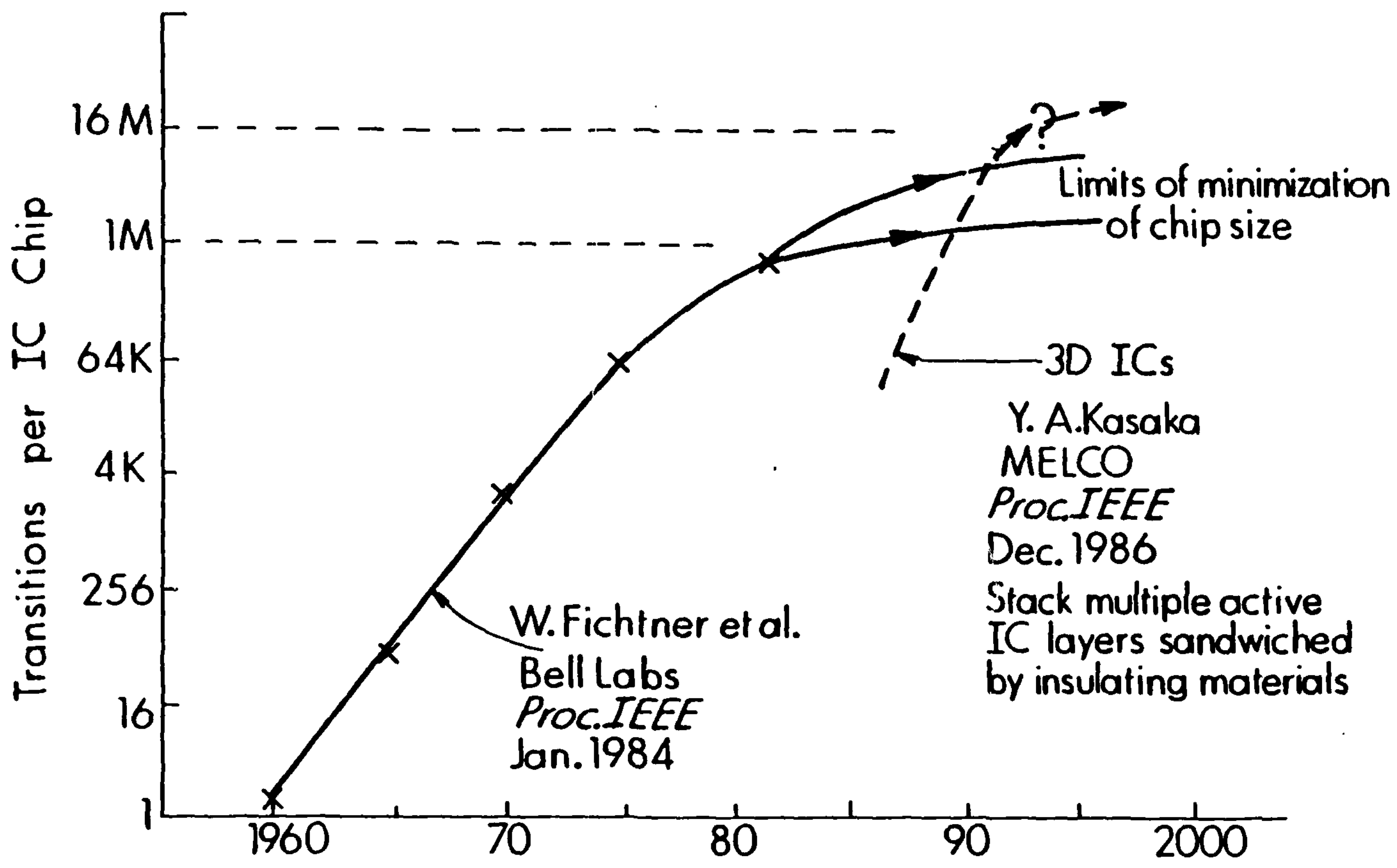


FIGURE 2

Management of **VLSI** Complexity-Complexity Barriers, Crises

VLSI: Growing to $>10^6$ Devices on an IC Chip

Successful Systems with over 10^6 Components

- Skyscrapers
- Wide Body Jets
- Nationwide Telephone Networks

Key is successful physical/logical partitioning and rigorous structuring

Similar Complexity Crisis was Faced in Software for Code of $\gg 10^4$ Lines

→ Greater Structure in Software, Teams of Software People rather than one "Wizard" must be *small* teams so communication between team members is manageable.

Key in **VLSI** is successful PARTITIONING into functions which can be compiled automatically or semiautomatically

See C.H. Sequin, UC-Berkeley, *Proc. IEEE*, Jan. 1983

FIGURE 3

VLSI Design Tool Limitations

Scftware Programmer and **VLSI** Designer Productivity

$\approx$ 10 Lines of Code/day – Surprisingly independent of language HOL or Assembly, but net efficiency varies greatly, e.g., assembler level LOAD vs some complex algebraic HOL inst.

ASIC **VLSI** Designer

$\approx$ 10 Items/day

Efficiency varies greatly whether it be

10 Processors or

10 Transistors/day

Thus Designer should deal with larger cells and automate the lower level design of these macrocells analogous to a HOL compiler

ASIC Design is 2, 3 Dimensional Problem

FIGURE 4

Selection of VLSI Design Methods

Overall Cost $C = \frac{C_{DEV}}{n} + C_{PROD}$, $C_{PROD} \sim$ Die Size

n = number of units produced

	Gate Array	Standard Cell	Full Custom Hand-Craft
$C_{PROD} \sim$ Die Size	1.0	0.75	0.4
$C_{DEV} \sim$ Dev Time	×	1.5 ×	6 × probably increases nonlinearly as gate count increases to large values

Ref 6k Gates, 2μm Technology
N. Okuda et al.
Toshiba
Proc. IEEE, Dec. 1986

FIGURE 5

Effectiveness of High Level Design Tools/ Silicon Compilers

	MACRO BLOCK CHIP DESIGN DENSITY (1.2 μm Design Rule)	
ROM	30,000 Transistors/mm^2	
RAM	10,000	
Regular Logic (e.g., ALUs)	3,000	
Random Logic	1,000	Low Efficiency

FIGURE 6

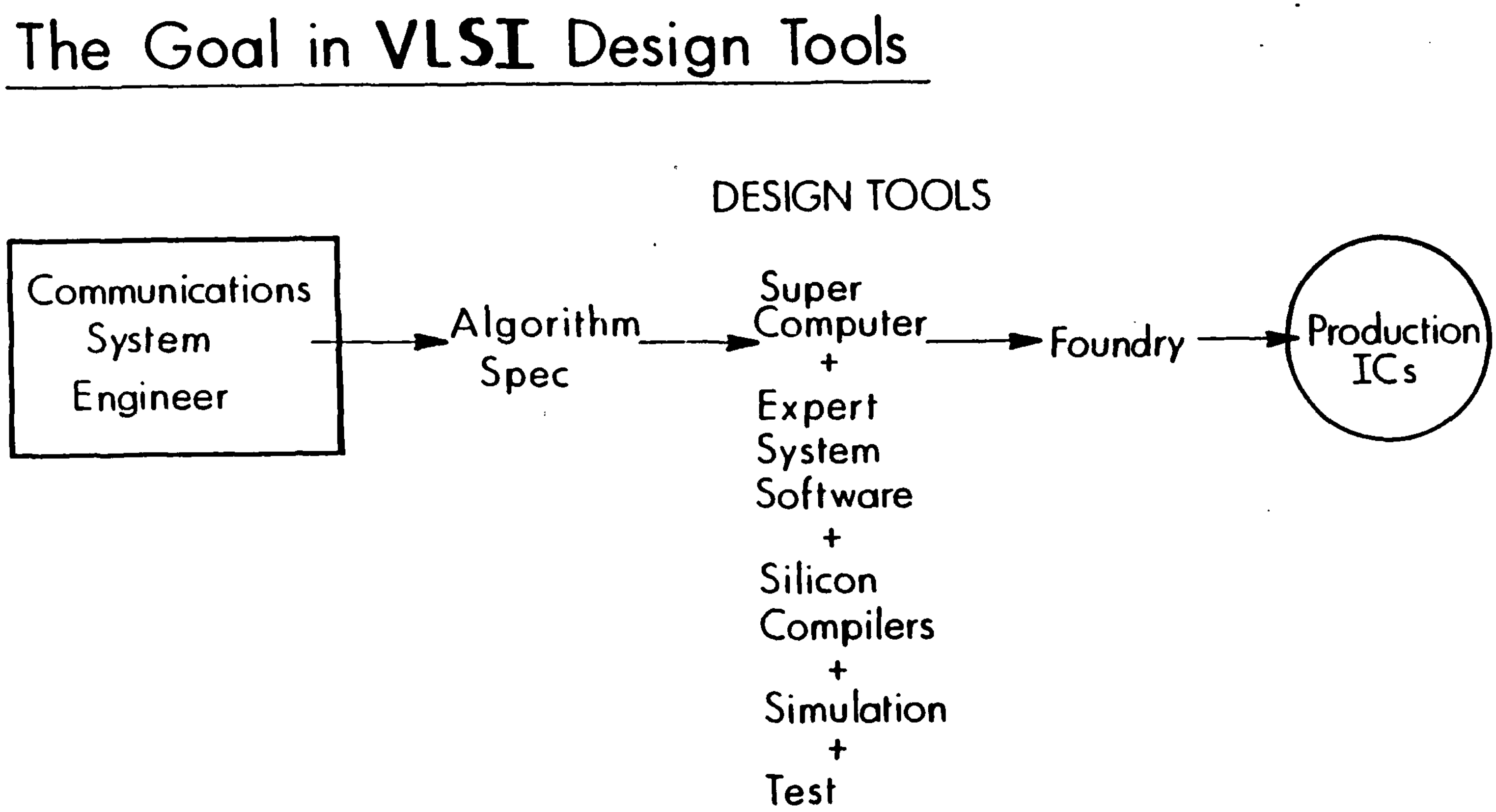

The Goal in VLSI Design Tools
DESIGN TOOLS
Communications System Engineer
Algorithm Spec
Super Computer
+
Expert System Software
+
Silicon Compilers
+
Simulation
+
Test
Foundry
Production ICs

FIGURE 7

VLSI Design Tools – Problems Now and the Future Hope

Complexity Management – Tools for *Partitioning*

CAD/CAE WorkStations & **VLSI** "FOUNDY" *Interaction*

Problem Now – Too many unique systems and fast moving technology
Need more uniformity of data formats – Compatible Formats
Different machines can then be used on same design

Silicon Compilers – HIGHLY EFFICIENT design of low level operations
Problem Now – Lack of efficiency except for MEM

AI – Expert Systems – Incorporate forms of Heuristics used by skilled human designers with rich data bases

SuperComputers as **VLSI** Design Tools – MiniSuperComputer Work stations SuperComputer & SuperAccelerator *Simulators*

VLSI Design Teams – Teams of Designers interacting with standard interfaces – Not just one "wizard," but still small teams Electronic Communication between Team Members

Self Testing & Self Healing **VLSI** Designs

FIGURE 8

workstations and VLSI foundries all working in parallel with little commonalty (see FIGURE 8). You have the people in the workstations business, like Daisy, Mentor Graphics, Apollo, Sun Microsystems, and so on, and then you have the various foundries, each of which has its own software tools. The interaction between these is not very good at this point. I think that we have a real problem because there are too many unique systems and in addition you have a very fast moving technology. It is not clear to me how this problem is all going to be resolved and I don't really have an answer for it. However, we definitely need more uniformity in the data formats so that systems can begin to talk better one to the other.

For the complexity management problem I described earlier, we need better tools for the partitioning of the VLSI chip into reasonable cells that can be put into a silicon compiler. In the silicon compiler area as I mentioned earlier there is a lack of efficiency, but I think a good many people are working on that issue, and I think that we're going to see some real progress in the next five years. Expert systems can incorporate some forms of the heuristics that are used by skilled human VLSI designers and then couple that with the rich databases that are available for the macro cells. As time moves on there will be more and more of those basic macro cell libraries available. Supercomputers and superaccelerators as VLSI design tools can make the VLSI circuits more and more powerful, and thus make the supercomputers themselves much less expensive and more powerful and hence more available as a VLSI design tool. As I mentioned earlier regarding VLSI design teams I think that our interest here is in making sure that we can operate efficiently with teams of designers, not just a wizard or two here and there. Then finally, one other thing that I think we'll see for these million or so gate count chips is that some of the chips will be self-healing, self-testing, or self-diagnostic. With a million or 5 million gates on one chip one would need to have redundancy built into the chip such that if a small microscopic fraction of the chip becomes disabled for some reason, that the whole thing doesn't have to be tossed in the garbage pit.

My view of the future of VLSI is a very optimistic one. I think that we're going to see design tools such that the communication system engineer in fact can really have much closer interaction with the VLSI foundry and much shorter design time. I see breadboarding as a way of life disappearing entirely. You really can't breadboard precisely any of these VLSI chips as most of you already know. Some of the prototyping that one had done in the past, I believe, will be a way of the past rather than of the future. Thank you.

BOOTON: Any questions?

CHETHIK: Well, perhaps more of a challenge than a question. One of the things that comes up for me on a number of occasions, I've been asked to quantify productivity which seems like an impossible task. From a software designer point of view, one can sit down and compute the number of lines of code per day that a software engineer produces, and as Jim mentioned whether these are written in assembly language or major code segments, his productivity can be measured somehow quantitatively. How does one go about quantifying this for hardware designers? I saw the first attempt at telling us that a VLSI designer can design ten elements a day, I think. I'd like to hear more about

that.

SPILKER: I don't really have a definition of productivity; I wish I did. The context in which I raised that was simply that if you look at the VLSI design productivity in the sense of how many gates can be efficiently designed on a VLSI chip per person per day, and when I say design I mean the whole process of all the way through the simulation and checking out the interfacing. So I think one measure is clearly how many gates can a designer produce per day. I wouldn't propose that as being the best, but it is certainly a measure.

BOOTON: I have one question for you, Jim. As things change in the direction that you've indicated, what does this say about the nature of the kind of person who should be the com-system engineer? At one time com-system engineers had to do no noise theory, modulation theory, and know how to do link budgets. And when you're coming to putting most of the system on one large chip, what's the nature of the kind of person who ought to be the com-system engineer?

SPILKER: I think that the com-system engineer, as time moves on, is in some sense going to be broader in his background. For example, the question of management of complexity of the systems. You know, I certainly never took any courses on complexity management at Stanford; in fact, I don't think there are many courses on complexity management in most schools even today. But nonetheless I think that's an important issue. The second one, of course, is getting much more familiar with the digital VLSI technology. I think that you have a number of universities now where the VLSI and semiconductor groups are over here in one portion of the university, and here are the communication system engineers in another portion, and over there are the computer science majors. It's remarkable how little interaction there is. The way we operate at our company is with highly projectized teams of engineers -- system engineers, communications system engineers, hardware designers, software designers working side by side in an effort to reduce the communication problem. However this mode of operation is considerably different than what you often see within the university campuses. I think that is one of the things that I would argue for, that students should be a little bit broader.

In the future you will be able to sit down as a, say Ray Pickholtz who's a super spread spectrum engineer, and design these fantastic communications systems and come up with a receiver design that can be translated right now into a hardware chip, and by golly, the damn thing actually works. And that's the amazing part to me, when you go through this complex sequence of operations, the chip actually works. I mean it's a revolution! I think that the interaction and broadening of the base of knowledge of our engineers is probably happening now to some extent, but not nearly to the extent that it should.

BOOTON: Thank you. Aside to my speakers, I'd like to pick up copies of your viewgraphs before you leave, so that USC can put these into the record of the conference, and we'll get them back to you. Alright, we have one more speaker today, George Turin from the University of California at Berkeley. Sorry to put you last, George, but ....

TURIN: Jim, one of the things that's worse than having your name start with "S" is having your name start with "T" ... [laughter] and I have a colleague at Berkeley who's Lofti Zadeh, you probably know

him .... [laughter]

I should probably start with an apology. I have been out of the research game for some years; I took a detour through academic administration. So some of the material is a little bit old and I think some of you in the audience have already seen it.

Chuck Weber of USC, over there, this morning said something very kind to me. He said, "George, welcome back to the research community," and I feel pretty good about that. But the downside of that, Chuck, is I'm a little bit rusty. Some of that administrative experience was at that other little university across town from USC. I can tell you, in private, some good USC-UCLA jokes. USC comes out the worst for the wear, Chuck, in those jokes . . .

At any rate, I'm going to talk a little bit about what I've called CADIPAC, Computer Aided Design of Packet Networks. The idea came to me seven or eight or nine years ago when I was with my then-teenage son, and he took me down to a Pacman parlor. I watched some of these kids, eleven or twelve years old, put a quarter into the Pacman machine and they would play that machine for hours without ever losing their quarter. I mean, they'd spend the quarter eventually but they'd win game after game after game after game before running out of the quarter, and I was fascinated by that. I'd put a quarter in and Pacman was eaten up by one of the monsters in 15 seconds. In pondering that, it occurred to me that the eye and the brain is a tremendous tool combined with some very good color graphics for doing things by experience. So I thought, "Instead of having these kids watching this little guy go around eating up dots, and then when the monsters turn another color watching out that Pacman doesn't get eaten up by monsters, what if I had 25 or 50 kids sitting at really fancy color graphics terminals, showing a display of a network, maybe a complicated network, maybe not 1,000 nodes but maybe on the order of 100 nodes. And what if each one of these kids would have as his responsibility to get a packet through the network, watching the other 50 or so packets being jockeyed through the network. He might see another packet heading for a node and think, 'Uh, oh -- that packet is going to go into a buffer and I'm going to get into that buffer after him and that's going to delay me, so maybe I better take a different route through some other nodes,' that sort of thing. And what if I let them play, these 25 or 50 kids, and give them points, scores and the whole bit, you know, maybe even make some money out of this to build a system." [laughter]

WELCH: ... maybe you should charge them quarters ....

TURIN: Yeah, we might actually do that because, as those in business know, what you charge is what you're valued at, so if you charge them zero they probably won't play.

At any rate the idea sounded interesting. The idea would be that these kids would get expert at jockeying packets through the network and then eventually when they got their end-to-end delay down to an average where it was some sort of stable number, you'd try to figure out what they did. What was their algorithm to get packets through the network? Or, maybe the computer could figure that out.

Now that's too hard a problem, so I figured I'd start on another problem, which would be to have simply a single screen which displayed a network, and to watch packets go through the network, according

to algorithms we'd designed ourselves. Now you'd want to see things that are much more dynamic than what you usually get out of simulation, because after all, this is a simulation system with a dynamic sort of quasi-real time display. You'd want to see transient events: you'd want to see the deadlocks occur, you'd want to see buffers overflowing, you'd want to see if a loop occurs. And you'd want to have a lot of fancy icons to show you those things happening, like if a buffer overflows you may actually see the packets spilling onto the floor. (We have a starburst in our system.) Or if a loop occurs, that should be highlighted; or if a link is getting up to capacity, you'd want to see that link glowing brighter and brighter red. Or, to show the general heat of the network, various areas of the network might be against various background shades of red.

The purpose would be that, if you design an algorithm and funny things start to happen, you'd be alerted to them. For example, let's say the only criterion is least end-to-end delay and maybe the algorithm therefore specifies going through the least number of links; that would tend to force packets through the center of the network, so the center of the network would get very hot and everybody gets crammed up there -- you want to see that sort of thing happening.

When we started building a system -- and this a little old (actually the students I had working for me on it were getting out of the pipeline at UC-Berkeley while I was at UCLA) -- we decided to make the overall architecture of CADIPAC modular (VIEWGRAPH #1). The modules in yellow are user programmable modules: these are nodes, and this is a channel. Everything else is a template into which these modules fit, for example, the event-queue manager that is shown.

CADIPAC sits on top of BLOSIM, which stands for Block Simulator, which sounds like it's very much like BOSS which was talked about earlier; it's a block oriented simulation system, and you can replicate blocks as many times as you want and change the parameters in the blocks. There's a packet generator, there are some display drivers -- there's a graphics display driver, there's a statistical display driver -- and there's an input console so the user can stop the motion on the screen as things occur. You can change the parameters, you can even change algorithms during the running of the simulation to see if you can affect the behavior of the network.

I don't want to say very much more about this; in fact, we're currently changing much of this architecture, we're going to make this a workstation-oriented program. (Right now CADIPAC runs on a VAX 750 with a dumb graphics display, but we're going to put it on a MicroVAX GPX over the summer.)

Let me show you some of the displays. The simple type of display is the statistics display (VIEWGRAPH #2). As the network evolves, one slows up the traffic introduced at various nodes and as the offered traffic goes up, one gets throughput and delay and buffer occupancy statistics. Nothing in this viewgraph is very fancy or very new. This is the type of thing that you would get out of a batch simulation.

Our hope though in this CADIPAC system is to see things that you don't see through mathematical analysis using queueing theory, which isn't usually powerful enough, or through batch simulation, where the problem is that you have tons of data coming out, lots of graphs, but you don't

get this intuitive insight into what's happening in the network. We'd like to see transients that analysis and batch simulation don't show, rare events that you might not anticipate in the system design. So the statistical displays on this viewgraph are not the interesting output.

The interesting output is the color-graphics-oriented display. We only have some very simple applications programmed now. Let me show you one network, a toy network (VIEWGRAPH #3). It's simply a ring network; you can't really see the links very well, but they're unit directional links. There are only six nodes, and the nodes are characterized by simply being buffers. Some very simple statistics show short-time averaged offered traffic, throughput, and average end-to-end delay. The traffic that's being offered in this particular toy network at each node is Poisson with the same rate of offered traffic at each node. In this particular case there's 1/3 of a packet on the average per time slot being offered at each node. The destination of any packet is uniformly distributed over the other five nodes, but it has to go around in a unidirectional manner, in this case counterclockwise. There's no flow control at the nodes, and if a packet gets to a buffer and the buffer is full, the packet simply drops on the floor, and (you can't see it here) when that happens there's a little starburst that occurs as one of the icons.

We're not very far along right now in terms of fanciness, but even this simple ring showed us a phenomenon that may or may not be not real. (That's another problem with simulation: sometimes you get phenomena that are simply artifacts of the random number sequence you put in, but they also may be a real phenomena.) We tended to see that when we got to steady state -- and the steady state here is an offered traffic of 2, throughput of 1.7 (the other .3 packets per slot being dropped on the floor because of an unavailable buffer) and average delay (meaning number of slots to get to the final destination) of 15.4 -- when you get to this steady state, you notice that often, not always but often, two adjacent buffers are full or almost full -- as seen in the two nodes on the lower right. And what's funny is that, as you keep on watching, they tend not always again, to precess around the network in the opposite direction from the flow of traffic. You know, that's kind of funny; you would expect it to happen if there were back pressure in the network, so that if a packet were introduced at a node and the next node were full and had to say "I can't accept any more packets," the first node's buffer would also start filling up. As the second node emptied out, the first's buffer would be filling up, but then the node behind it would have to start filling up too, and you'd start getting that sort of precessing phenomenon. But here we didn't have any flow control, we were dropping packets on the floor.

Now this may not be a real phenomenon -- I've watched this simulation run with the same random number seed for the clock running up to thousands, and it seems to be happening. That's the sort of intuitive insight into the behavior of a network that I think this type of visual orientation, color-graphics orientation, can give you that other tools can't give you.

We did other types of networks. This (VIEWGRAPH #4) is a linked-cluster network, where you have a backbone -- in this case just a loop -- and clusters of nodes attached to servers that were on the backbone. We actually ran only long-term statistics for this sort of network, in order to compare to queueing theory results with

CADIPAC statistics. Queueing theory, even for simple merges of data streams or splits of data streams becomes very complicated, so we had to make some assumptions about the nature of the statistics to do the queueing theory. Then the question became whether the assumptions are supportable or not, and we ran CADIPAC to show that the results from CADIPAC were much same as the queueing theory. This was a standard queueing simulation used to verify theory.

Again, our thrust with CADIPAC is not mainly statistical output. CADIPAC is mostly supposed to be a heuristic tool to help you understand what is really going on in a network, so that maybe you can understand queueing theory and the design process better.

What we hope to do over the summer is to port CADIPAC, which is now running on a rather antiquated AD767 display connected to a 750 VAX through a 10 kilobaud line to a MicroVAX GPX which has very good graphics. It'll have a couple of workstations, which will stand alone, it'll have its own disks and so forth, and so several students will be able to work with it at the same time. We're planning to do a lot more with the icons of the system; for example, some of the things I mentioned about the color of the links as they go up to capacity, and so forth. Also what I want to do is make it much more user friendly, so that a node may not be characterized just by a buffer -- that's a very simple node, but there might be much more complicated nodes. A node might just be a dot, but you'll be able to put the cursor, by using a mouse, at that dot and zoom in, and be able to look inside the node, and there might be several buffers in it. For example, if the computer says something funny is happening at a node -- you know, packets are getting jammed up in there -- you might say, okay, let's zoom in on that node, and let's see the buffers and everything else in that node and see what's happening in that node. I want more user-friendly capability so that you could put the cursor at the node and have a window open up and you'll be able to easily modify the parameters of the node. Right now we have to do this by writing code, but what I want to do is enable the user just to say, "Okay, here's a node, I want to change it," get a window and then start changing the node on the screen.

As I mentioned at the outset, the material I showed you is pretty old, and my hopes haven't happened yet, so I have to apologize for a halfway-house presentation. But we're excited by the possibilities. Thank you.

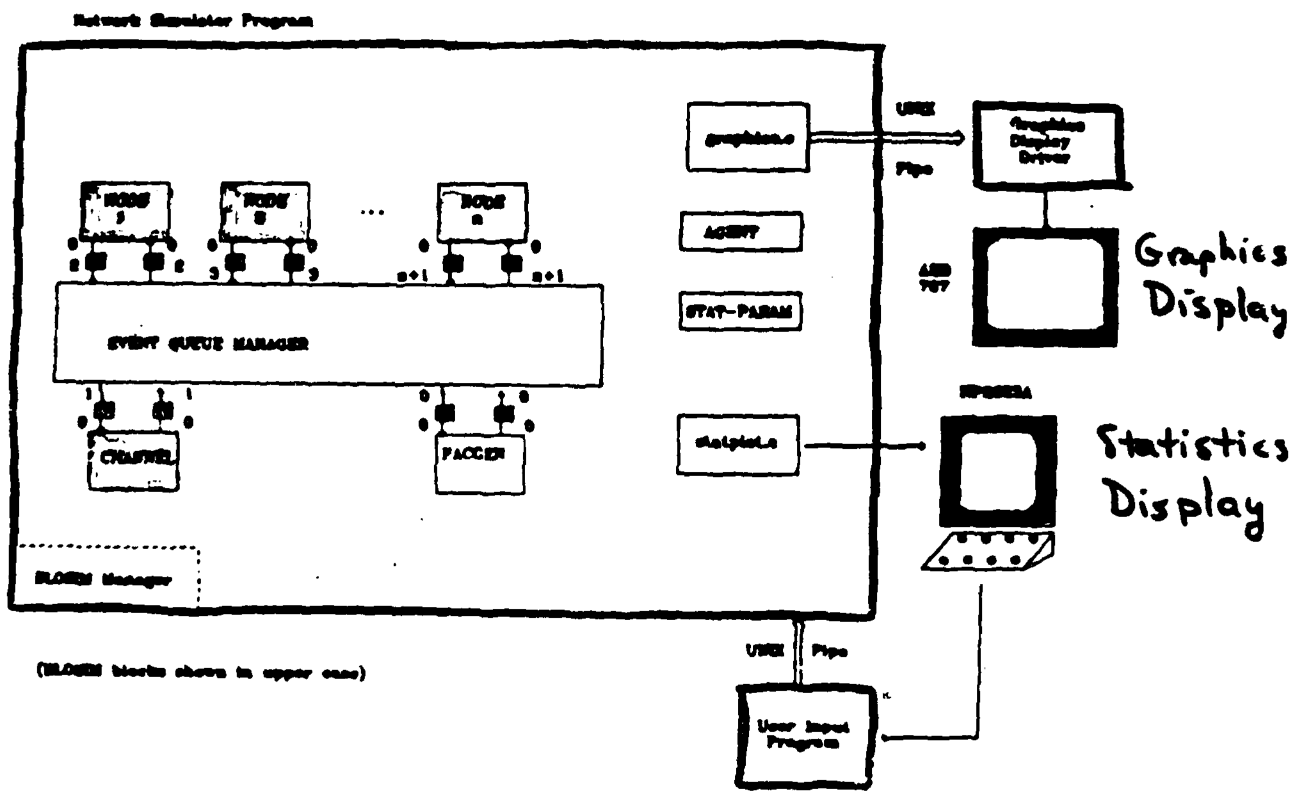
BOOTON: Thank you, George, for a very interesting finish to this morning's session. I want to thank all the speakers for taking part in this, and thank the audience for your questions and involvement, and your patience with us. We're minus 15 minutes now, as far as reserve. Any more questions? Yes ....

MOHANTY: Any more architecture for token bus? Do you have any restriction how many nodes you can take or is it unlimited how many nodes you can take in the token bus?

TURIN: That wasn't a token bus itself, it was a toy bus, is what I'd say. There was no control whatsoever for flow, there was no organization, you just jammed packets into each node and they had to somehow work their way -- if they didn't get through the network, they disappeared. But we can structure virtually any topology.

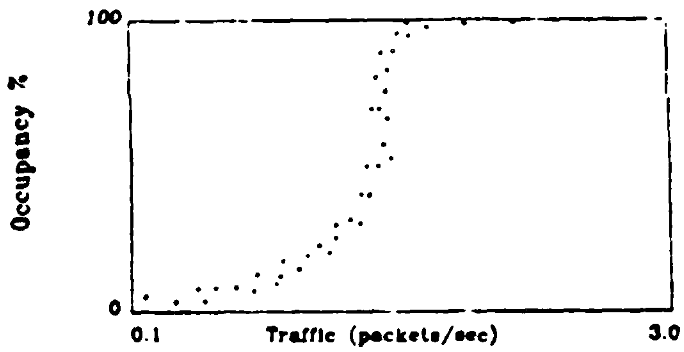
Another thing that we're going to do is allow this to have mobile networks. I'm very interested in mobile packet networks.

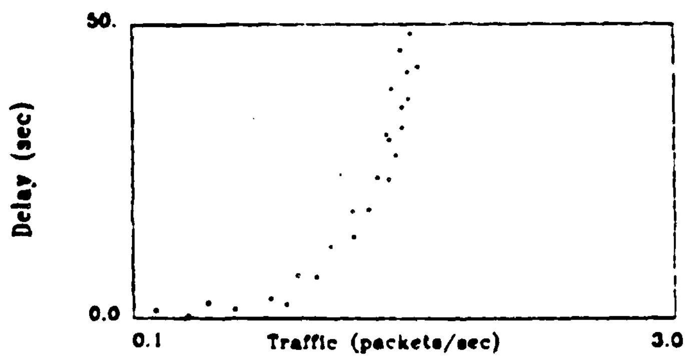
CADIPAC Architecture

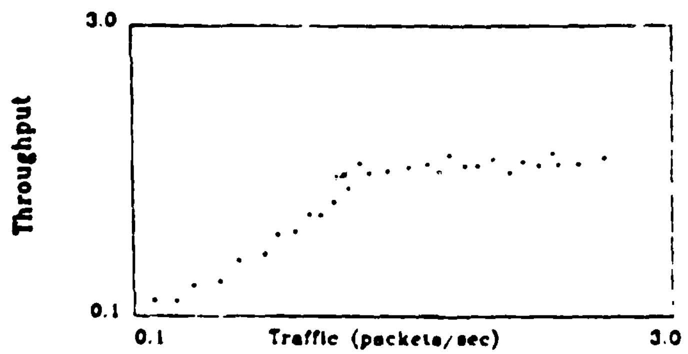


VIEWGRAPH #1

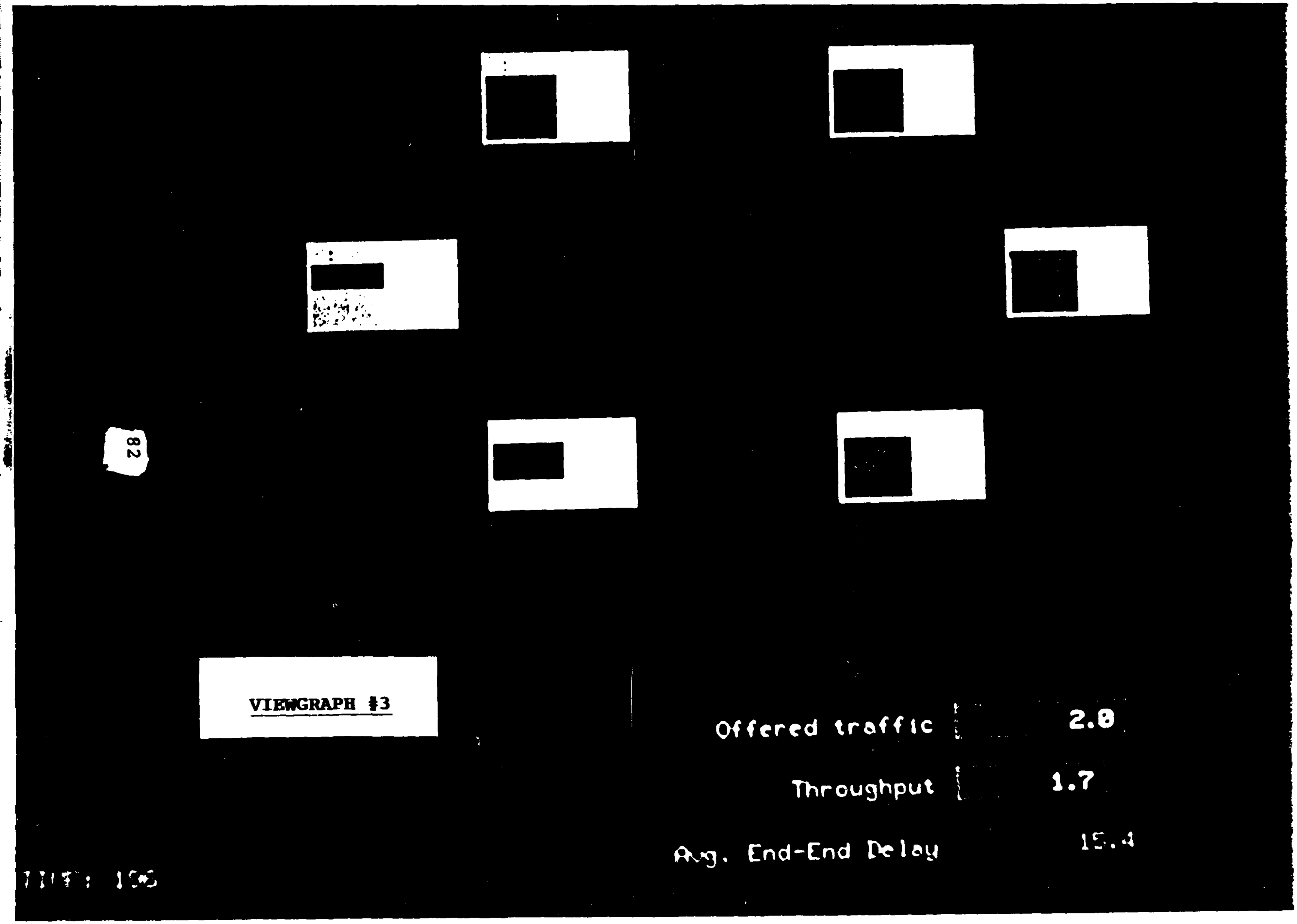
Statistics Display

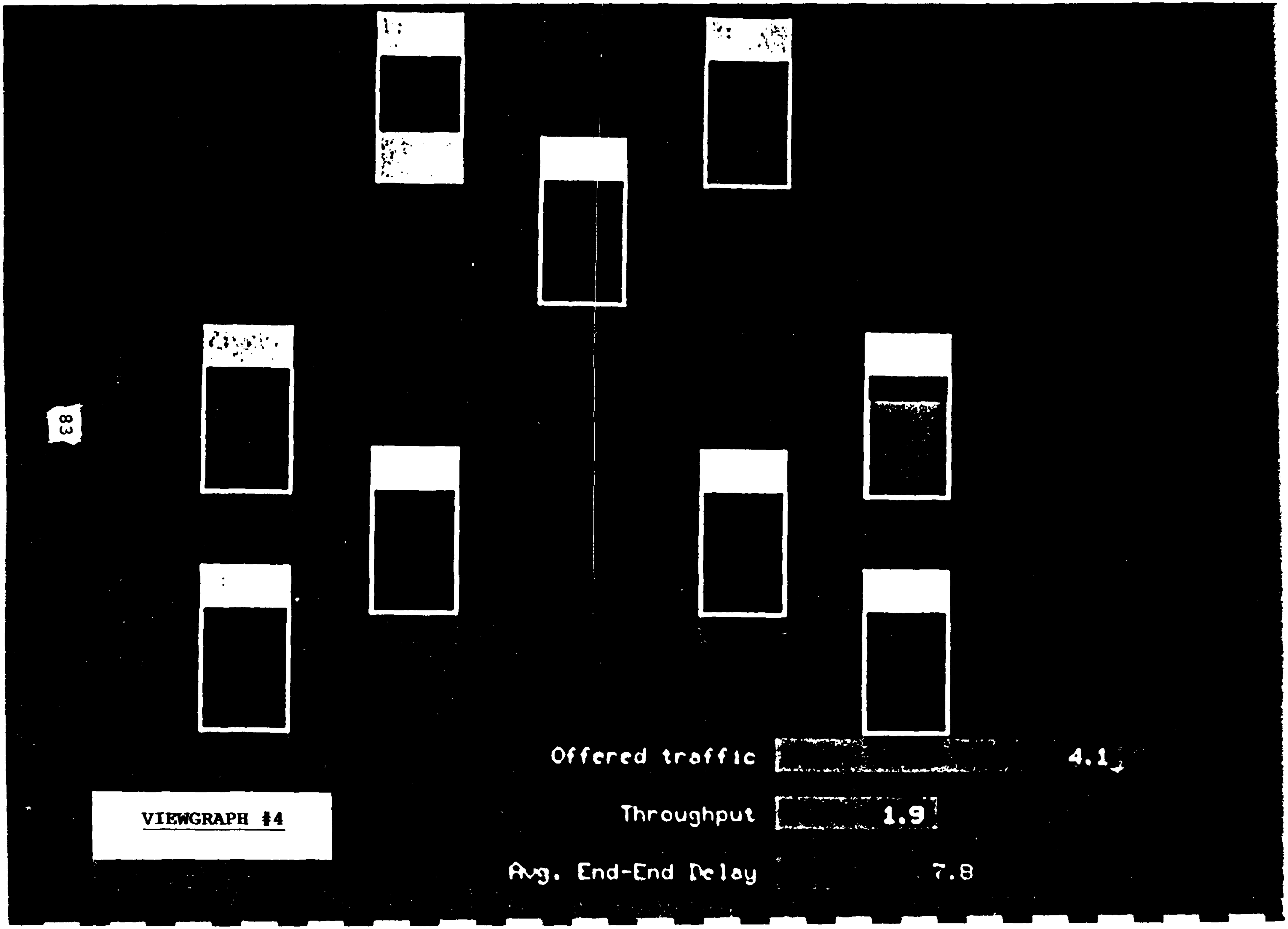






VIEWGRAPH #2


VIEWGRAPH #3
Offered traffic 2.8
Throughput 1.7
Avg. End-End Delay 15.4



VIEWGRAPH #4

and so we're also going to let the nodes move around and let the links make and break, and then try to figure out protocols for dynamic reorganization of the network as links make and break because of shadowing or fading, as new users come in, as nodes move around, and see how those protocols for reorganization work. Again this may not be a final design tool. The idea here is a heuristic tool to enable you to understand things that I don't think are very well understood right now. As far as what we have now, we have just what's up here [laughter] and that's not a token bus.

MOHANTY: Thank you.

BOOTON: Thank you. Yes, one more question ....

KUROSE: Concerning what you have up here, I was wondering if you've given some thought about how you might visually deal with the problem of scaling, so if I had a packet radio network with 1,000 nodes in that, how to deal with that visually?

TURIN: Well the MicroVAX GPX, I think, has about one million pixels, so I can't have more than a million nodes. I think that perceptually if we start displaying more than about 25-50 nodes at any one time, there's just too much information to encompass, unless I get some pretty bright teenage kids to sit down with quarters. So we expect if we do do something bigger than that, we'll have a zoom capability. Just as we're able to zoom into the innards of a node, we would be able to window a particular section of the network and look only at the behavior of that and imagine that all the rest of the network is actually in the outside world driving what we're looking at. So we have a lot of good ideas, I think they're good, and we're just going to scale this up full speed if we can, starting this summer.

SANDER: You are talking about displaying on this graphics global information a network where apparently you haven't thought about any size limitation, but there will obviously be some overhead to getting that global information about the network into a central location to display on such graphics. Have you thought about that problem?

TURIN: We're going to run -- you know, it's like any other simulation system or number crunching system -- we're going to run out of CPU cycles and disk and ability to get this thing to run reasonably fast as we increase the size of the network. We don't have a good sizing right now. Again my guess, from having played around with this a little bit over the past 4 or 5 years, is that when we get up to 25-50 nodes displayed with any degree of real activity, and if we try to run it in a more or less quasi-real time way, we're going to start running into trouble, especially if we're working on a workstation. My hope is that workstation technology is going to advance at least as fast as we can work on this thing, so that CPU speeds and fast memory and so forth are all going to grow accordingly.

My real concern is not so much how fast the CPU can run and how fast we can run the simulator, but is how much we can display before we overload the eye-brain capability to comprehend what's going on. The Pacman syndrome is what I would call it. So again, I'm not trying to promise that this becomes a real large-scale network design tool, but more a heuristic design tool for the network designer to get good ideas about what's good or bad about particular algorithms or protocols and be able then to modify them.

This hopefully will go hand-in-hand also with the emulation system at Berkeley that was just mentioned. Varaiya and Walrand are doing a sort of mixed hardware-software emulation system and they don't have any good display capability -- well, they have some -- but hopefully we'll be able to link this up with the emulator. If we can replace the simulator with an emulator, then maybe we can make CADIPAC go a lot faster. The limitation in my mind is the ability of the eye and the brain to comprehend what's going on.

SANDER: I guess this question may not be appropriate if you think about using this tool simply as a simulator, but such a tool would be invaluable in a real network, particularly in a military environment where you have so much dynamics. But the question, the original question I really intended to address, was if overhead on the network itself is getting information about distant nodes and all the nodes in the network back to a central maintenance control station, that we might have such a display as you're talking about.

TURIN: Yeah, I would hope that at least what we've learned would be helpful for the real-time display of actual networks that are operating, but that's not our goal in this particular project. It seems to me almost a sure thing that what we come up with would be helpful in that problem, which is the display of actually operating real-time networks with many, many nodes, and we'll be able to pan and zoom and so forth, and look at various parts of the network.

BOOTON: Thank you again, George. Bob, do you want to make any comments?

SCHOLTZ: I think this is a great opening panel. We've covered a lot of different ideas, which will probably pop up again in later panels in more detail. I hope that later panels somehow will be able to find a block of time near the end for some real discussion of issues that will get the audience into it a little bit more. It may take another half day or so to reach that form of operation. Let's shoot for that if we can. I'd like to thank all the panelists. I thought you gave us a great deal of data and some new ideas to work on. Thank you very much.

USC-CSI WORKSHOP ON ADVANCED COMMUNICATION SYSTEM ENGINEERING

Proceedings of Session Two: Computer-Aided Modeling, Analysis, and Design of Communication Systems

SHANMUGAN: The title of this session is "Computer-Aided Modeling, Analysis and Design of Communications" including Communication Links and Networks. An alternate and perhaps more appropriate title could be, "CAD Tools for Advanced Communications Systems Engineering." Whereas the preceding session this morning focused on future CAD tools that are yet to be developed, our session will focus on CAD tools that we have now and CAD tools that will be available in the near future.

My name is Sam Shanmugan and I am with the University of Kansas. The panelists for the session are Phil Balaban (AT&T Bell Labs), Mike Jeruchim (General Electric), Frank Amoroso (Hughes Aircraft), Hussein Mouftah (Queens University, Canada), Jim Kurose (University of Massachusetts), and A.R.K. Sastry (Rockwell Science Center). Mike, Phil, Frank and I will discuss CAD tools at the link level. Hussein, Sastry and Jim will address issues related to CAD tools for networks.

Systems engineers are responsible for comparative evaluation of competing designs. This is often done in the form of trade-off studies and comparisons using a variety of computer-aided analysis and simulations. If the models and simulation techniques are valid and accurate, then comparative performance evaluation will produce meaningful results. Of course, simulation and analysis will have to be validated in parts with experimental data.

In many situations, simulations and model based analysis results and experimental data do not agree within reasonable bounds. This is due to problems in modeling methodologies, simulation techniques and input data. These issues will be addressed by our panelists this afternoon.

At the systems engineering level, we need good CAD tools to reduce the time required to produce and validate new designs (rapid prototyping) and to reduce the time required to insert new technologies into products. CAD tools can also improve the accuracy of designs and reduce the cost of designs for high quality products that are produced in small lot sizes. Good tools can also be used to create a corporate-wide "knowledge" data base that can capture and transfer design expertise from project to project.

We use a variety of CAD/CAE tools. During the early (research) stages of a product development we rely on mathematical modeling and analysis techniques and use these techniques for proofs of a concept. A considerable amount of effort is currently devoted towards developing additional mathematical modeling and analysis techniques.

During the design and development phase of a product, we use a variety of computer-aided modeling analysis and design tools. The focus of our session will be on these tools.

If we take a hierarchical view of CAD tools for advanced communication systems engineering, we start with the network layer

COMPUTER-AIDED MODELING ANALYSIS AND DESIGN OF COMMUNICATION SYSTEMS (LINKS & NETWORKS)

PANELISTS

K. SAM SHANMUGAN
University of Kansas, Lawrence, KS

MIKE JERUCHIM
General Electric, Philadelphia, PA

PHIL BALABAN
AT&T Bell Labs, Holmdel, NJ

FRANK AMOROSO
Hughes Aircraft, Fullerton, CA

JIM KUROSE
University of Massachusetts, Amherst, MA

HUSSEIN MOUFTAH
Queen's University at Kingston, Ontario, Canada

A.R.K. SASTRY
Rockwell Science Center, Thousand Oaks, CA

SESSION OUTLINE

1. Introduction - CAAD Tools for Transmission Systems (Sam Shanmugan)

2. Systems Modeling and Simulation Techniques (Mike Jeruchim)

3. Channel Modeling (Frank Amoroso)

4. Role of Expert Systems (Phil Balaban)

---

5. CAAD Tools for Networks (A.R.K. Sastry)

6. Current State of the Art (Jim Kurose)

7. Role of Expert Systems (Hussein Mouftah)

8. Wrap Up (Sam Shanmugan)

BOSS: Block Oriented Systems Simulator
(Overview and Demonstration)

at the top. At this level, we are dealing with network protocols, routing algorithms, congestion control, etc., and the performance measures are delays and network throughput. The next level in the hierarchy is the communications link which is made up of functional blocks such as encoders/decoders, modulators/demodulators and channels, and the performance of the link is measured in terms of error probabilities and signal to noise ratios. The third level in the hierarchy is circuits: microwave and electronic. Power levels, impedance matching, and speed are some issues of concern at this level.

While we have standard CAD/CAM tools for circuit analysis, design and manufacturing (SPICE and MOSIS for example), we do not have any standard tools for analysis and design at the link and network level. At the link level, computer-aided analysis and design tools are reaching a level of maturity and it is expected that software packages such as BOSS will be widely used for link analysis and design in the near future. Standard tools for network analysis and design are expected to evolve during the next five to ten years.

At the present time, there is a lack of integration between the CAD tools used as various levels. The interface between these levels is performed manually. Integration of CAD tools and interfacing the CAD tools to hardware databases are current areas of research. Improving the computational efficiency and accuracy of models and simulation techniques, and developing an expert systems framework for CAD tools are other areas of current research.

The panel discussion this afternoon will cover the following topics:

(a) *Transmission Systems (Links)*

Modeling and Simulation Techniques - Mike Jeruchim
Channel Modeling - Frank Amoroso
Expert Systems Framework for Transmission Systems - Phil Balaban

(b) *Networks*

Current State of CAD Tools for Networks - Jim Kurose and A.R.K. Sastry
Expert Systems Framework - Hussein Mouftah

We were planning to demonstrate the Block-oriented Systems Simulator at the end of the discussion this afternoon. But, because of hardware problems, we will not be able to do that today.

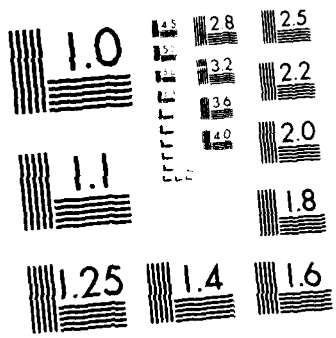
With that quick introduction, I would like to turn the floor over to Mike Jeruchim now.

JERUCHIM: I notice we have a crowded agenda, so Mr. Chairman, I want to inform you that my presentation is totally modular. You can cut me off in mid sentence and you won't have lost a thing. [laughter]

When Sam invited me to be on his panel, I must confess I was somewhat worried. He told me that this was to be a workshop on "Advanced" Communication Systems, and "advanced" work always seems to be something that others are doing, but not oneself. So I was somewhat comforted by this morning's presentations, which confirmed my notions of what "advanced" is. In fact, by splicing 3-4 of the presentations this morning you'll have perhaps most of what I have to say.

Sam asked me to talk about the current state-of-the-art of simulation techniques within the context of computer-aided modeling, analysis, and design of communication systems. Let me first remark that

AD-A194 336 ADVANCED COMMUNICATION SYSTEM ENGINEERING PROCEEDINGS 2/5
WORKSHOP HELD IN SE..(U) UNIVERSITY OF SOUTHERN
CALIFORNIA LOS ANGELES COMMUNICATION S.. R A SCHOLTZ
UNCLASSIFIED MAR 88 CSI-88-03-02 ARO-24913.1-EL-CF F/G 25/5 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 196[illegible] A

this long label, "computer-aided modeling, analysis, and design" is essentially a synonym for what most of us call simulation. So, for brevity and without risk of confusion, I will confine myself to the use of the shorter term. Secondly, although it will be implicit from the context, I want to make it clear that I will be addressing only link design issues and not networking issues.

Now, the phrase "state-of-the-art" is one of those terms that can be subjectively interpreted, almost at will, to suit one's purposes. It does have a connotation of ground-breaking work, but the utility of that work may or may not be far-reaching. In any case, as you can see from [VIEWGRAPH #1], defining what is state-of-the-art in simulation may be problematical. This is because simulation itself is a multifaceted discipline, as you can see from the viewgraph, and state-of-the-art activity may be taking place on many fronts.

I intended actually just to brush by a few things that I consider state-of-the-art activities in simulation techniques, and I want to concentrate, rather, on an aspect of simulation which is not explicit in this chart [VIEWGRAPH #1]; and that is, the synthesizing aspect of the various components of simulation that are shown. The synthesis that I'm referring to is consonant with the systems orientation of this workshop, and it is a system engineering, or system design methodology built around simulation as the basic performance prediction tool.

Now, as to state-of-the-art simulation techniques, the next viewgraph [VIEWGRAPH #2] shows a short list of what I consider to be significant developments that are in various stages of progress. As I implied earlier, and illustrated by the sketch on this viewgraph, what constitutes the state-of-the-art, like beauty, is in the eyes of the beholder. So, this list of activities coincides with the kinds of needs that are surfacing where I work, and may not fit everyone's idea of the state-of-the-art.

The first activity listed on this chart [VIEWGRAPH #2] concerns the simulation structure itself. With respect to the simulation structure, the ideal would be an expert-assisted, user-friendly, graphics-oriented interface structure. We are working on this sort of thing, as are various organizations. As far as anything that is actually operational, BOSS comes as close to achieving these kinds of desiderata as anything I know of. And since you've already hear about BOSS, and will hear more from Sam, I'll say no more about it.

Another area in which state-of-the-art work is being done or needs to be done is *in estimation techniques. We are in an era* where customers are requiring more and more accuracy from transmitted data, and bit-error-rate requirements of the order of 10^{-6} to 10^{-8} are now surfacing. No matter how good your models are, if you were to try to estimate that kind of performance with reliability, you'd have to keep a major mainframe running continuously for literally 2-3 months. So we need sophisticated ways to extrapolate error-rate performance measures. Although there has been a fair amount of activity in this regard, over the last few years, I would say that there is still room for useful contributions. Right now, the "hot" topic in this area appears to be importance sampling, and the research being done may yet yield a generally useful and applicable extrapolation procedure.

Another aspect which we are beginning to see a need for, under the heading of "simulation techniques", is multi-rate sampling. This is something which tends to be

done naturally in hardware where we are often faced with situations where a system has many signals coming in on different lines with widely different data rates.

Of course, we may well have to deal with the same situation in software. Typically, in this situation the traditional approach has been to sample all signals at the highest rate necessary for any signal in the system. This is potentially wasteful of computation because low data rate signals would be sampled as rapidly as the highest rate signal. This morning, someone remarked (facetiously) that if you add up many billions of dollars, you can get into *real* money! Well, if you add up many billions of microseconds, you can get into *real* amounts of time. The point is, as a system becomes increasingly large and complex, you can get yourself in a situation where you can use up inordinate amounts of time if you have not taken advantage of every available technique that can make computation more efficient. Multi-rate sampling is one technique for contributing to that efficiency. The basic idea is that signals of different bandwidths in different parts of the systems are sampled at the rate appropriate for their bandwidths, and no faster. The necessary conversion, decimation, and interpolation techniques are relatively well known, but have not really been applied to simulation contexts (to my knowledge). So, this is another of the state-of-the-art simulation techniques that people are working in.

One of the areas that is emerging, and where we need to improve simulation techniques, is on links where we have mixed analog and digital modulation schemes. The challenge here, for example, is to relate digital link performance parameters to the analog signal performance parameters. An example of the kind of link that I'm referring to, is where you may have a number of analog signals coming in to a node of some kind, say in FDMA fashion, and then you "bundle" all of these signals together, digitize the group and transmit it digitally.

In the modeling area, there is no doubt much activity going on that would qualify as "state-of-the-art". I would describe state-of-the-art work in this context as something which yields not only a more accurate model of a device or process, but also a model which is efficient to simulate. On this viewgraph [VIEWGRAPH #2] I have listed just three representative items, which also happen to coincide with some of the work that we are doing. The first item, the modeling of non-linear amplifiers with memory, in particular TWT's, is an area which has received only a small amount of attention. This is an area where, I think, communication engineers and physicists have to get together and cooperate. Most of the models that we generally see have been developed by communication engineers, based on the externals of the box, and here's one area where we may need to get into the internals of the box.

Next on the list is something that I find we need in our work, generally, namely good models for non-Gaussian noise processes, and in particular, for the one I've listed here, phase noise. We also find, as we try to refine the modeling more and more, that we can no longer model A/D converters as ideal devices. Their temporal characteristics now have to be modeled in some way. They are not just instantaneous samplers but they have some kind of time constant and so on.

I know that I have dealt with the items on this chart [VIEWGRAPH #2] rather summarily. However, as I said earlier, I want to address most of my remarks to the

last item on the chart, namely system engineering methodology and the role of simulation in that activity. In the little time that I have, I will try to give you a flavor of that interaction. I do want to say that the perspective which I bring to this, is that of an organization which designs and builds space systems. Some of our views and procedures may not be totally applicable to all cases, but I do believe that they are, if the basic goal is to provide for a minimum level of performance of a fairly complex system over a relatively long period of time. This is essentially the message of the first bullet on the next chart [VIEWGRAPH #3]. The system design procedure starts with recognizing that the system owner wants a guaranteed performance. He doesn't care if it's better than that, if it doesn't cost him extra money, but he does want performance to be no worse than a *certain standard*, for a long period of time, and over all the possible conditions that the system can encounter. The other reality is that excessive margin is increasingly expensive. You can fill in your own number here next to M, (in the second bullet), but it is megadollars per dB, and you can't just throw margins around anymore in a cost-conscious environment. The obvious conclusion is that we need as accurate an estimation of system performance as we can get. There's a problem, which perhaps is not fully recognized, and this is the tendency to think of a given system as an entity having completely definable characteristics. In reality we do not know precisely what these characteristics are. In fact, the system behavior is a "moving target." Its transfer functions vary with time, both in a long-term sense, as in "aging", and in a short-term sense, which is typically cyclical; for example, temperature variations. A system also generally has redundant chains of equipment and so, potentially, we have a multiplicity of systems within the single entity that we call *the* system. You cannot really control an animal of that nature, other than through a relatively simplified control structure, and this structure is basically built around specifications. In order for this approach to work and to be practical, the specifications must not be too numerous, they must not unreasonably constrain design trades, but yet they must provide a reasonable assurance that the performance objective will be met. The way we deal with this situation is the following: we synthesize a so-called "spec system," i.e., a communication system which attempts to "look" like the real system, but meets every single specification at its outer limit, so that it (the spec system) presumably performs as poorly as it is ever possible for the real system to perform.

As indicated on the next viewgraph [VIEWGRAPH #4], the primary tool for determining specs and performance is simulation. The simulation must be capable of tracking the system from inception to realization. We use it to develop specifications, to verify performance, and if specifications on a part of the system are not met at one or another stage of the system development, we can plug the measured characteristics back into the simulation to see whether a waiver should be granted or not on that particular piece of equipment. The simulation should have a flexible block synthesis using combinations of parameter specifications, measured values, and theoretical-analytical models. We want the block synthesis to be ideally as flexible as it is for your hand to draw an arbitrary curve with a pencil on a piece of paper. Of course, as I said earlier, the simulation should be as accurate as possible, and based

on experience, I believe that the state-of-the-art for complex satellite systems simulation is approximately half a dB, either way, of the truth.

In this context, I would like to make an observation, namely that the truth is unknowable, and this is merely a statement of the uncertainty principle. There is a tendency to think of a measured result as "the truth". In fact, measured results are not unambiguous, because our ability to set and calibrate equipment has an inherent variability which is basically irreducible. This was brought home to me many years ago, shortly after we had first developed our digital communication simulation. We embarked on a demonstration to "validate" the simulation. Equipment was assembled in a laboratory, and the various relevant characteristics were measured so that these characteristics could be inserted into the simulation. The physical equipment and the simulation were exercised separately and independently and we came out with BER curves that could be compared. I was surprised (at that time) upon examining the measured data to see that there was in fact a variability to it. A number of BER curves had been measured over a period of days and by different technicians, all highly qualified. The various curves did not all agree with one another. This basically was due to the fact that there is an uncertainty in the E_b/N_0 calibration, and this uncertainty is on the order of 0.5 dB. There is very little variability in the BER measurement itself since one will usually observe a sufficiently large number of errors. In simulation, on the other hand, one can in effect calibrate the E_b/N_0 axis perfectly, but one can usually not afford to spend the computer time necessary to observe a large number of errors. So, here we have variability in the vertical, rather than the horizontal, axis. The basic point is that there is inherent uncertainty in both measurement and simulation, and that, in a sense, each can be used to verify the other.

There's a set of tradeoffs for what an ideal simulation should do, as I indicate on the next viewgraph [VIEWGRAPH #5]. And it may be paradoxical that an ideal simulation is not necessarily the best replication or emulation of the physical system as they may be, because of the interaction with people. People have to interact with the simulator. It takes time to develop understandable models. We have, say, 15 or 20 young engineers who live most of their lives in what I call prison; they call it the terminal room! They have to interact with this simulation constantly, and they have to be able to understand the simulation and its models without being experts on either. Runtime, as I mentioned earlier is very important. For a complex system, literally hundreds or thousands of iterations may be needed when you're doing trade-off analyses. And you can't wait two or three days for each run. So there's also a trade between how accurate a model is and how fast it is. It's almost given that the more accurate a model is, the longer the run time will be. That's not necessarily the case where you have a poor model; sometimes it's just a matter of the right equations. It takes just as much time or as little time to solve a good equation as a bad equation, but generally speaking it takes longer to run a model with lots of "knobs" on it. Well, we have now completed what's on this chart.

Part of what I just said has an effect on how you develop a model to begin with. There are different approaches to developing models, and these are shown here on the next chart [VIEWGRAPH #6]. These are the major modeling approaches that I

can think of. You can try to do exactly what the device is doing, down to almost the atomic level, or if this is too detailed a level, down to the component level. You don't necessarily need to have a model that is close to the physics of a device. But in some cases, for example the traveling wave tube model that I mentioned earlier, you may have to be at least one level deeper in detail than it is possible to infer strictly from the standard communication characterization of the device. At the other end of the simplicity scale, there is what I call the phenomelogical model. Here, you don't try to replicate or emulate what the box is physically doing, but only what its ultimate effect is. So, for example, if you're talking about a bit synchronizer, say, you don't need to have models for all the little pieces and boxes which comprise this device. As far as the system aspect is concerned, the only thing the system cares about is how regular the sampling is. If you can somehow abstract a sampling process that has an appropriate kind of jitter associated with it, you have reduced the complexity of this model by an order of magnitude. Similar modeling considerations also apply to the simulation as a whole. Here, I distinguish two kinds of approaches that I call statistical-time versus statistical-time/ensemble. By statistical-time, I mean that you are generating or observing the evolution in time of a process. However, if you are observing slow processes in time, you cannot often afford to wait for the process to have had a statistically significant realization. If you're talking about a slow process like a bit sync jitter of PLL phase error, these processes are slow relative to data rate, and you can't wait long enough to have the slow processes take on all of their representative values. So, in that case you may know an ensemble property, like a pdf. Then, you may be able to run a short statistical-time simulation, and then average some parameter over an ensemble distribution.

The substance of the last four viewgraphs is capsuled graphically in the next viewgraph [VIEWGRAPH #7]. This viewgraph is really a flow diagram of the system design process, and it can be seen that simulation has a central place in this process. Its importance lies in the fact that it is our performance evaluation tool, and essentially replaces what used to be done by "analysis". Of course, the simulation must be used intelligently, meaning that the inputs to it must be meaningful. Thus, for example, we have to carefully synthesize a spec model, which represents the outside chance that the system reaches all of its spec limits. I have labelled this spec system a "pedigreed" system, on this chart, because its characteristics are not arbitrarily determined, but, rather, inferred from measurements, history, and experience on similar systems. So, the pedigreed system represents what you might guess the real system would be like, if it went to the extremes of all the specification limits.

You'll notice on the chart that I have validation in two place. One is what I call validation of the simulation, and this is basically a direct check of the simulation against measurement. It ensures, for example, that our models are reasonably correct and that there are no software bugs. Then there is something that is labeled "system validation", which is synonymous with link closure. In space communication, link closure is paramount. That is, the link budget must provide the necessary E_b/N_0 (or SNR) to support the desired BER performance. Note that since the necessary E_b/N_0 for the BER in question is that which applies for the spec system, it is not possible to

GENERAL ELECTRIC

COMPUTER-AIDED MODELING, ANALYSIS AND DESIGN OF COMMUNICATION SYSTEMS

SYSTEM INTEGRATION DEPARTMENT

- PERFORMANCE MEASURES
  - BER
  - SNR
  - ERRORS
  - CONFIDENCE
  - HISTOGRAMS
  - WAVEFORMS
  - SPECTRA
  - SIGNAL SPACE PLOTS
  - SCATTER PLOTS
  - EYE DIAGRAMS
  - ACQUISITION TIMES
  - CORRELATION LEVELS
- PERFORMANCE EVALUATION TECHNIQUES
  - MONTE CARLO
  - QUASI-ANALYTIC
  - EXTREME-VALUE STATISTICS
  - TAIL EXTRAPOLATION
  - IMPORTANCE SAMPLING
  - HYBRID MONTE CARLO/ANALYSIS
  - HYBRID SOFTWARE/HARDWARE
  - HYBRID SOFTWARE/FIRMWARE
- DEVICE/SUBSYSTEM MODELING
  - FILTERS (IIR, FIR)
  - ANALOG/DIGITAL FILTER APPROX
  - A/D - D/A
  - MODULATORS
  - DEMODULATORS
  - ENCODERS
  - DECODERS
  - CARRIER SYNCHRONIZATION
  - BIT SYNCHRONIZATION
  - NONLINEAR AMPLIFIERS
  - NONLINEARITIES
  - INTERPOLATION
  - DECIMATION
  - MIXERS/FREQUENCY CONVERTERS
  - OSCILLATORS
  - OPTICAL FIBERS
  - LASERS
  - LIGHT DETECTORS
  - ATMOSPHERE
- COMMUNICATION THEORY
  - MODULATION
  - COMPLEX ENVELOPE
  - SAMPLING
  - FOURIER THEORY
  - DFT
  - Z-TRANSFORM
  - DSP
  - CODING/DECODING
  - SPECTRAL ESTIMATION
  - PROBABILITY/RANDOM PROCESSES
  - EQUALIZATION
  - FADING/DISPERSIVE MEDIA
  - NOISE (THERMAL IMPULSIVE, MAN MADE)
  - INTERFERENCE
- SOFTWARE/COMPUTATIONAL TECHNIQUES
  - PROGRAMMING LANGUAGES
  - PROGRAMMING TECHNIQUES
  - USER INTERFACE
  - GRAPHICS
  - PSEUDO-RANDOM NUMBER GENERATION
  - FFT
  - NUMERICAL ALGORITHM

VIEWGRAPH #1

GENERAL ELECTRIC

SOME SELECTED STATE-OF-THE-ART DEVELOPMENTS IN PROGRESS

SYSTEM INTEGRATION DEPARTMENT

- SIMULATION STRUCTURE
  - EXPERT-ASSISTED, USER-FRIENDLY, GRAPHICS INTERFACE
- BER ESTIMATION/EXTRAPOLATION TECHNIQUE
  - IMPORTANCE SAMPLING FOR NONLINEAR/LONG MEMORY CHANNELS
- SIMULATION TECHNIQUES
  - MULTIRATE SAMPLING
  - MIXED ANALOG/DIGITAL TRANSMISSION SCHEMES
- MODELING
  - NONLINEAR AMPLIFIERS WITH MEMORY
  - EFFICIENT AND FLEXIBLE OSCILLATOR PHASE NOISE GENERATOR
  - REALISTIC A/D-D/A MODELING
- SYSTEM ENGINEERING METHODOLOGY

VIEWGRAPH #2

GENERAL ELECTRIC

SYSTEM ENGINEERING ISSUES WITH RESPECT TO COMMUNICATION SYSTEM SIMULATION

SYSTEM INTEGRATION DEPARTMENT

- THE SYSTEM OWNER WANTS A GUARANTEED PERFORMANCE OVER SPECIFIED CONDITIONS
  - LIFE
  - ENVIRONMENTAL VARIATIONS
- EXCESSIVE MARGIN IS INCREASINGLY EXPENSIVE ($ M/DB) AND INCREASINGLY UNACCEPTABLE
- THEREFORE ACCURATE AS POSSIBLE ESTIMATION OF SYSTEM PERFORMANCE IS ESSENTIAL
- SYSTEM CHARACTERISTICS (TRANSFER FUNCTIONS, ETC.) ARE INHERENTLY FUZZY, AS A FUNCTION OF TIME
- SPECIFICATIONS ARE THE CONTROL FOR ASSURING PERFORMANCE
- THE COMMUNICATION SYSTEM ENGINEER SYNTHESIZES A "SPEC" SYSTEM BASED ON SATISFYING SPECIFICATIONS
- THE SPEC SYSTEM EVOLVES OVER TIME WITH INCREASING KNOWLEDGE OF SYSTEM

VIEWGRAPH #3

GENERAL ELECTRIC

SYSTEM ENGINEERING ISSUES WITH RESPECT TO COMMUNICATION SYSTEM SIMULATION (CONT)

SYSTEM INTEGRATION DEPARTMENT

- THE PRIMARY TOOL FOR DETERMINING SPEC SYSTEM PERFORMANCE IS SIMULATION
- THE SIMULATION MUST BE CAPABLE OF "TRACKING" THE SYSTEM FROM INCEPTION TO REALIZATION
  - DEVELOP SPECIFICATIONS
  - VERIFY PERFORMANCE
  - WAIVER PROCESS
- THE SIMULATION SHOULD HAVE FLEXIBLE BLOCK SYNTHESIS USING COMBINATIONS OF:
  - PARAMETER SPECIFICATIONS
  - MEASURED VALUES
  - THEORETICAL AND ANALYTICAL MODELS
- THE SIMULATION SHOULD BE AS ACCURATE AS POSSIBLE: STATE-OF-THE-ART FOR A COMPLEX NONLINEAR SYSTEM IS WITHIN 0.5 DB OF MEASURED VALUES

VIEWGRAPH #4

GENERAL ELECTRIC

SYSTEM ENGINEERING ISSUES WITH RESPECT TO COMMUNICATION SYSTEM SIMULATION (CONT)

- THERE IS A SET OF TRADE-OFFS FOR SYSTEM SIMULATION
  - ACCURACY IS RELATIVE TO THE INTENDED APPLICATION
  - RUNTIME IS IMPORTANT
  - EASE OF USE IS HIGHLY DESIRABLE IF NOT MANDATORY
  - HIGH-LEVEL VS LOW-LEVEL MODELING
- CHOICE OF MODELING TECHNIQUE HAS MAJOR IMPLICATIONS ON:
  - DEVELOPMENT TIME
  - EASE OF USE
  - GENERALITY
  - NON-EXPERT MULTI-USER APPLICATION

<u>VIEWGRAPH #5</u>

SYSTEM ENGINEERING ISSUES WITH RESPECT TO COMMUNICATION SYSTEM SIMULATION (CONT)

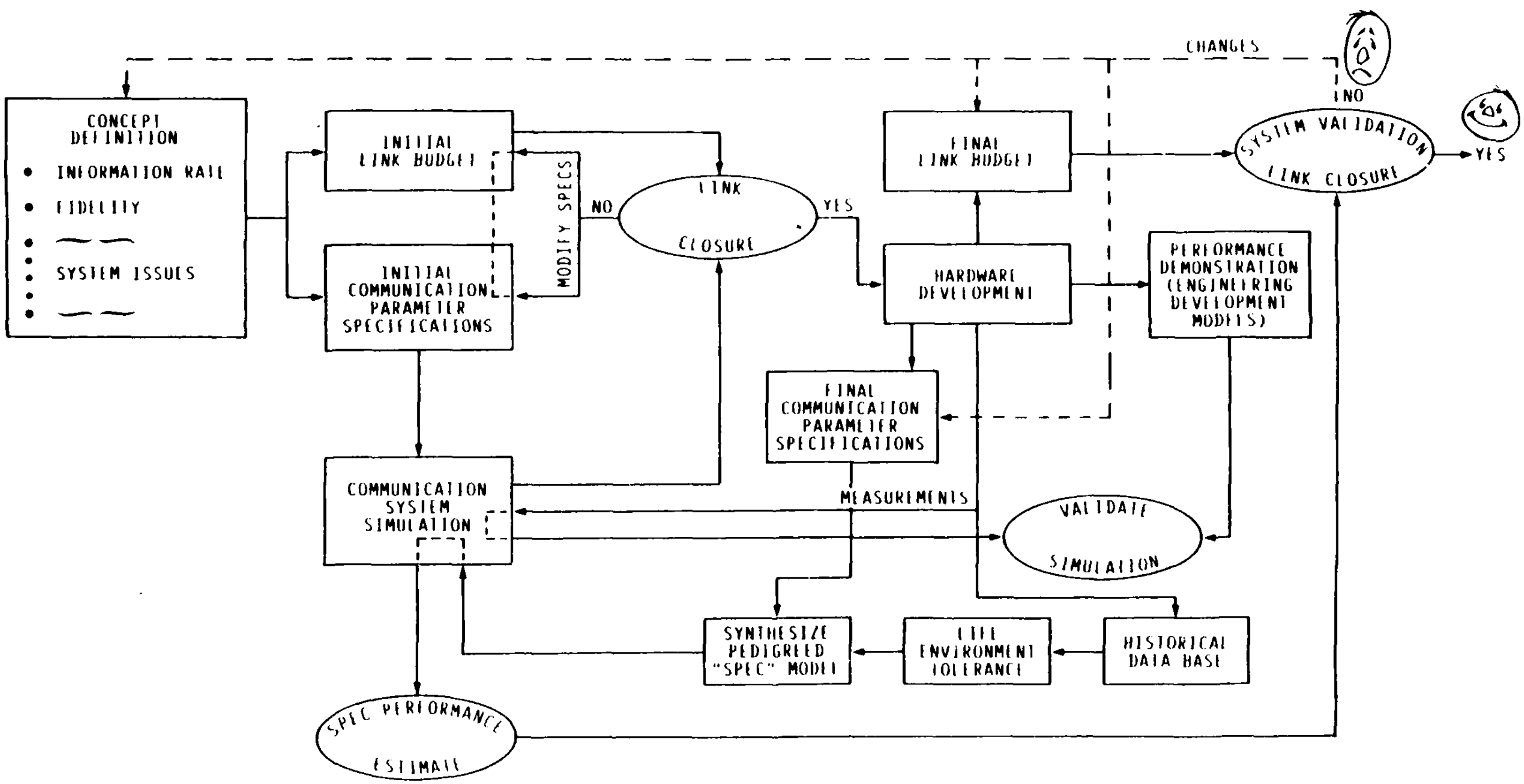
- MAJOR MODELING APPROACHES FOR DEVICES AND PROCESSES
  - EXPLICIT REPRESENTATION, OR "PHYSICAL" MODEL
    - ONE-TO-ONE MAPPING OF ACTUAL OPERATION, POSSIBLY WITH SOME APPROXIMATIONS FOR REDUCED COMPLEXITY
  - TRANSFER FUNCTION REPRESENTATION, OR "MATHEMATICAL" MODEL
    - MAY MAP THE PHYSICAL PROCESS
    - MAY BE AN ANALYTICAL EXPRESSION
    - MAY BE A TABLE LOOK-UP
    - MAY BE ALGORITHMIC
    - MAY REFLECT ACTUAL MEASUREMENTS
  - PHENOMENOLOGICAL MODEL
    - REPRESENTATION OF NET EFFECT AT POINT OF APPLICATION
  - STATISTICAL-TIME
    - EVOLUTION OF PROCESS IN TIME
  - STATISTICAL-TIME/ENSEMBLE
    - EVOLUTION OF PROCESS IN TIME COMBINED WITH AVERAGING OVER ENSEMBLE

VIEWGRAPH #6

GENERAL ELECTRIC

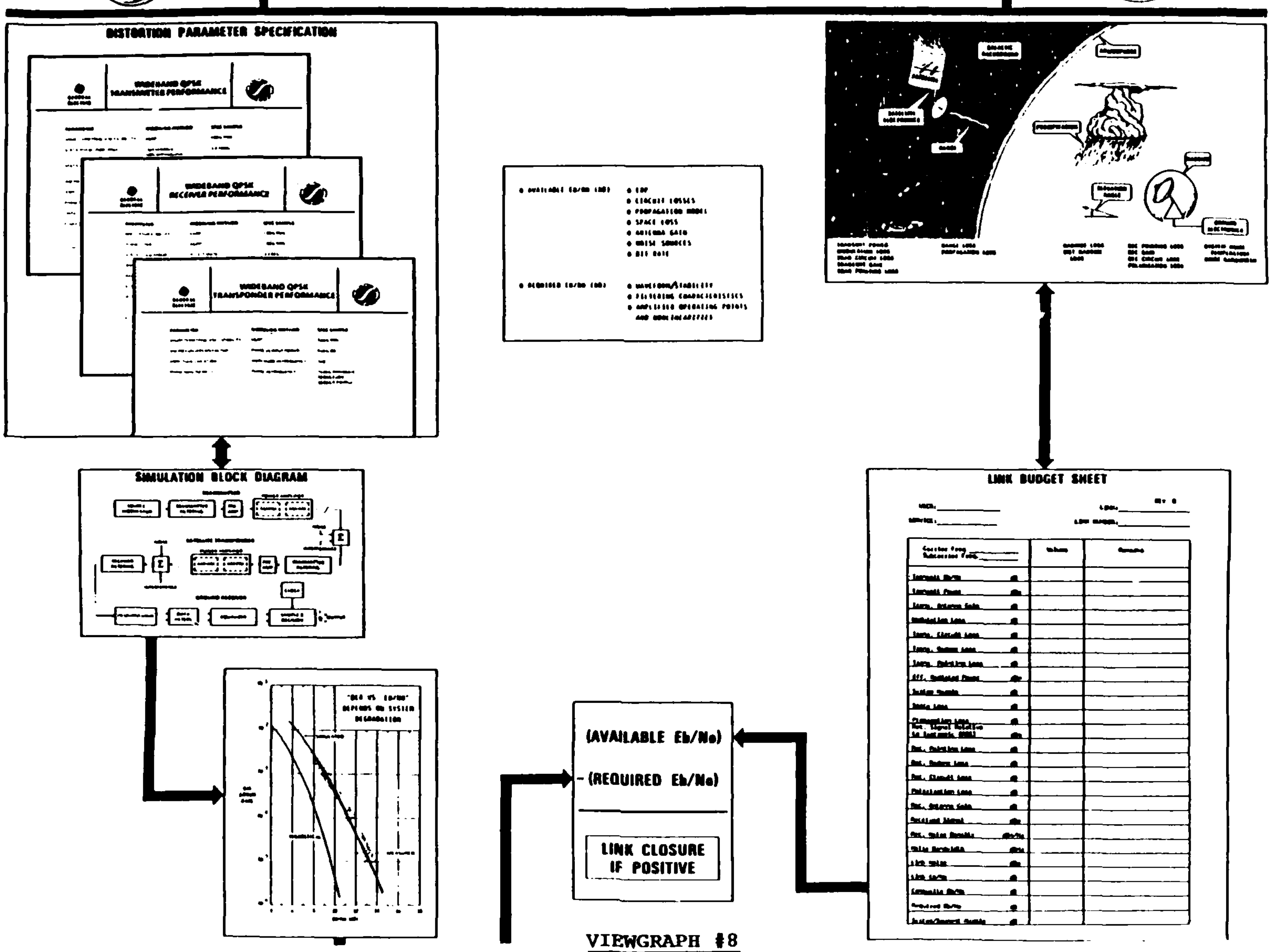
SYSTEM DESIGN METHODOLOGY AND ROLE OF COMMUNICATION SYSTEM SIMULATION

SYSTEM INTEGRATION DEPARTMENT



VIEWGRAPH #7

LINK CLOSURE: WHEN AVAILABLE Eb/No $\geq$ REQUIRED Eb/No



VIEWGRAPH #8

directly validate the system, unlike the simulation itself. Our objective is to obtain link closure with only modest margin. This is because, by its very nature, there in inherent margin in our methodology since it makes use of the "spec system", which represents the simultaneous realization of all specification limits. This, of course, is extremely unlikely to occur in practice.

I have just a few seconds left, so let me briefly flash the next viewgraph [VIEWGRAPH #8]. This is a specific embodiment for digital satellite systems, of the general methodology shown on the previous chart. Here, the world (unlike Gaul) is divided into two parts: the link budget part on the right-hand side, and the performance and distortion-related part on the left. The left-hand side tells us how much E_b/N_0 we *need* for a specified BER, and the right-hand side tells us how much E_b/N_0 is *available* from the nominally designed system. The difference between the two is margin. If margin is positive, we're OK, as long as it's not *too* positive. Otherwise, we have to iterate the design process until we do get an acceptable non-negative margin. This, in brief, is the process that we use for space system design. The remainder of my viewgraphs, which I will not present, are basically expansions of this theme. I see that I have run out of time, Mr. Chairman, so I will hand the floor back to you. Thank you all.

SHANMUGAN: This is a very good summary of systems engineering and the role of simulation and CAD tools for communications systems engineering. Do we have any questions for Mike? If not, we will go on to the next speaker. The next speaker is Frank Amoroso from Hughes Aircraft and he is going to talk about some issues on channel modeling and so forth.

AMOROSO: In this talk I wish to get deeply into the details of certain modeling issues. This is [SLIDE 2] the dense scatterer mobile environment. Probably the best way to say what that means is to show a picture. There are lots of scatterers around the mobile, and the objective is to communicate with this mobile. This visual is drawn from a mobile sat industry briefing at JPL in 1985. I think it is a good diagram; it encompasses the full range of possibilities. They happen to show a geostationary satellite transmitter, so the dense scatters could be mountains, trees, whatever. There's a diffuse component of the arriving signal which represents, in general, a collection of reflections from various places, through trees, possibly off many scatterers. In a minute I'll show an example of that diffuseness. Also, a "specular" component might exist, which goes from the transmitter directly to the antenna and sometimes is called the direct wave. These additional reflected components can sometimes be considered coherent, if the reflection is very simple as opposed to being diffused. This particular component is called a specular ground reflection. Therefore a great variety of path types enter the so-called dense scatterer mobile environment.

The transmitter into a mobile environment need not necessarily be a geosynchronous satellite. It can also be, as in this picture [SLIDE 3] from Bill Lee's book on Communications Engineering, a terrestrial base station, and the mobile (user) can be a car somewhere in a dense urban environment. This starts to introduce some of the characteristic parameters often found in problems of this sort. For example, let's say that a radio frequency impulse were transmitted from the base station. It arrives via various paths. In this diagram, there is

no specular component, only the diffuse one, since the signal arrives via many small reflectors. So what arrives at the mobile is a series of replicas of that transmitted impulse, showing various amplitudes and time delays depending on the lengths of those paths and the nature of the scatterers. Other parameters depend on the envelope, (a complex pulse envelope), not represented completely here. This illustration shows only the amplitude, $E(t)$, of a complex function of time. So there's a mean delay D, which is just a straightforward calculation of the first moment of $E(t)$. If this $E(t)$ is normalized to unit area, as if it were a probability density function, then there can be a measure of dispersion, much like the standard deviation of a probability density function. It is called **Delay Spread,** Δ . Mean time delays and delay spreads in various urban and suburban environments range from 1.3 down to 0.5 *microseconds*. In fact, $E(t)$ changes as the mobile moves, so $E(t)$ is too simplistic an expression. The $E(t)$ should have two time variables to indicate a time-varying complex function of time.

With this background of parameter characterization these [SLIDE 4] phenomena are often mentioned. Under multipath, there is a fading with motion. For example, transmit an unmodulated CW carrier and you would see that the received signal faded with motion in a complex way. Selective fading with frequency exists, so that if you just keep the mobile still and sweep the frequency of the transmitter, you'd see selective fading. If a narrow pulse were transmitted it will obviously arrive quite distorted. If you transmit an unmodulated CW carrier and the mobile is moving with appreciable velocity then you don't see pure CW, but an interesting spectrum. That depends on lots of conditions which I'll review briefly. Then there's something called **Shadowing.** If there are a few trees around and you go behind a tree, you find that the average signal level decreases. This is called Shadowing (you're in the shadow of a tree). Then there is the specular component, which would be a line of sight path, more or less, in addition to scatters. Furthermore, something else called mean excess propagation loss exists. In other words, if you can statistically characterize the fading with motion, due to multipath, this fading will carry some mean attenuation, which tends to be somewhat predictable. That attenuation would be the mean excess propagation loss, a kind of fudge factor which gets you to the correct mean signal strength, and believe me it is a very empirical thing. There have been published papers with empirical data on mean excess propagation loss.

So here [SLIDE 5] are the characterizations. For multipath, I showed a received pulsed power profile which changes as the vehicle moves. So for every different vehicle position, and this is motion on the orders of a wavelength, the pulse power profile changes. (You can also speak of this pulse power profile as a real non-negative function of time.) The complex pulse envelope also changes versus position, and the pulse power profile is just the modulus of that envelope function. For example, for an unmodulated CW or very narrow band signal, the received power fluctuates sometimes wildly as the vehicle moves. The fluctuation is usually Rayleigh or Rician. These statistics can be used to characterize the channel. There are applications for which those statistics alone are sufficient, and applications for which they are not. For example, you might want to take just a swept frequency response. This will also change with position (or time if the vehicle

is moving with a constant speed)!

The one parameter often connected with this is called **Correlation bandwidth,** the bandwidth over which you can keep signal phase coherent. Then the Doppler, as I mentioned before, has a power spectral density. Even if the transmitted wave is unmodulated CW, you will get something very interesting at the receiver. This spectrum is often used to characterize channels. Shadowing, of course is very often given with lognormal statistics, and you have to know the fade rate. Regarding the specular components, it's often important to know the ratio of specular power to mean power coming in from scatters. That's the Rician k factor, and this mean excess propagation loss, as I said, is mostly empirical.

Here are some examples which are taken from various places [SLIDE 6] which I've striven to identify faithfully. If you want to look back at them, that's a measured pulse power envelope somewhere in New York City. [UPPER LEFT] It spreads out over many microseconds and that was a delay of .69 microseconds and a delay spread of .77 microseconds. Now, the Fourier transform of the pulse power envelope looks like this. [LOWER LEFT] It obviously has something to do with correlation bandwidth, and in fact very often it is said that when you go down to the 50% point of this Fourier transform, that would be called the correlation bandwidth, in this case .751 MHz. This is a way of getting an idea of the bandwidth over which a signal can be held, phase-coherent. Obviously this correlation bandwidth is related to delay spread, we hope in some inverse relationship. In fact, he measured correlation bandwidth versus delay spread here [RIGHT] and we have an inverse relationship as suggested here by these straight lines. This becomes critical if the signal is going to have appreciable bandwidth, comparable to or greater than the the correlation bandwidth. The delay spread, of course, can change over decades of range, depending on where you are in the city, or where you are within the city block.

This [SLIDE 7] now relates to the statistics of the received signal when an unmodulated carrier was transmitted. Here the vehicle is rolling down an interstate highway in Texas, (measurements are in wavelengths), 20-40-60 wavelengths. So within orders of a wavelength, it passed behind a clump of trees here [LEFT]. This is signal blockage from a pine forest. He also showed cumulative distribution functions of the received signal [RIGHT]. This is the statistical characterization of that wiggly thing at various elevation angles. Since the transmitter was basically a balloon they (the testers) were able to change the elevation angle from the mobile to the transmitter. Of course at lower elevation angles, he's really looking through the trees, and that would be like Rayleigh fading with no specular component. Then at higher elevation angles he begins to show some evidence of an increasing specular component coming in, presumably over the tree tops, until you get into something that looks like higher Rician k factors.

Here is the spectrum [SLIDE 8] (if the vehicle is moving along and it happens to be within a perfect circle of uniformly distributed scatters. This result is much more general than I'm showing, but you can think about it more easily if you imagine scatters in a perfect circle. As the vehicle moves and an unmodulated CW carrier is transmitted, you have a maximum doppler frequency of V/λ and that's what the spectrum of the incoming signal looks like. This sort of function is a first derivative of an arc cosine so what started off as an

unmodulated carrier has been spread to something which is bowl-shaped and gets very high out at frequencies equal to the carrier plus and minus the maximum doppler frequency. That's for an omni-antenna, and for a directional antenna you get different spectra. There are people trying to demodulate data at rather high rates who worry about this.

Here is a model [SLIDE 9] that accounts partially for the phenomena. For this model we have Gaussian sources coming into I and Q channels and multiplying the signals and so on. That will simulate that effect of fading, which will give you this spectrum if you transmit an unmodulated carrier from the RF source. It's often used, looks pretty clever but doesn't tell you anything about delay spread. It doesn't give you any information about correlation bandwidth. Those (details) are gone. The assumption is that the transmitted signal is very narrowband. I've added some more pencil work here to show that shadowing could have been added, but it wasn't present in the original. This is an example of a model quite well conceived, but limited in the scope of what it can do for you, what it can show you.

We played around with one wherein a signal comes in from distant source and reflects off a bunch of point scatters [SLIDE 10]. Each one is an omni-directional scatterer; $1/(R^2)$ is the propagation loss to an antenna which can be permitted to move. This one happened to have a delay spread of 40 nanoseconds. Of course, simple rescaling of this would give whatever delay spread you want. I called this Constellation 1. Next [SLIDE 11] is Constellation 2, with a delay spread of 200 nanoseconds. So this [SLIDE 12] gives an honest-to-gosh pulse power envelope and can be used to investigate the effect of delay spread, especially when the transmitted signal is wideband [SLIDE 13]. In this case, we just tried it with an unmodulated carrier [SLIDE 14] and we did see fading that looked Rayleigh with respect to position changes. That's a span of 4 meters with this frequency at 1547 MHz. Both Constellation 1 at 40 nanoseconds, and Constellation 2, at 200 nanoseconds showed honest-to-gosh Rayleigh-like fading. When we added a specular component Rician $k=5$, that looked pretty good too. So, we compiled statistics [SLIDE 15]. We felt that we had validated the model.

This is what the spectrum [SLIDE 16] of the incoming signal looks like when you transmit an unmodulated carrier and you let the vehicle roll at 60 miles an hour. It should look bowl-shaped with the big peaks here. It's only a finite number of scatterers, and that was a specular component coming in at $+V/\lambda$. So it can be messy if we relate it to a specific situation.

Now the thing I wanted to show, just skip a couple of slides, is the swept frequency response [SLIDE 19]. We are going to hit this thing with a 10 MHz wideband pseudonoise data stream, and of course that's much wider than the correlation bandwidth. But there is a reason for doing this. What we are trying to do is to find out whether we could get rid of the Rayleigh fading by transmitting a very wideband signal that spanned over the frequency response. In fact, the individual chips in the direct sequence pseudonoise wideband signals were so narrow in time that we could resolve the arriving paths.

So here it is [SLIDE 20]. This [TOP] is with an unmodulated carrier where you get deep Rayleigh fades. We do the same antenna track, the same transmitter power, but with a 10 megabit per second [MID-

HUGHES
AIRCRAFT COMPANY

ISSUES IN MODELING THE DENSE SCATTERER MOBILE ENVIRONMENT*

FRANK AMOROSO

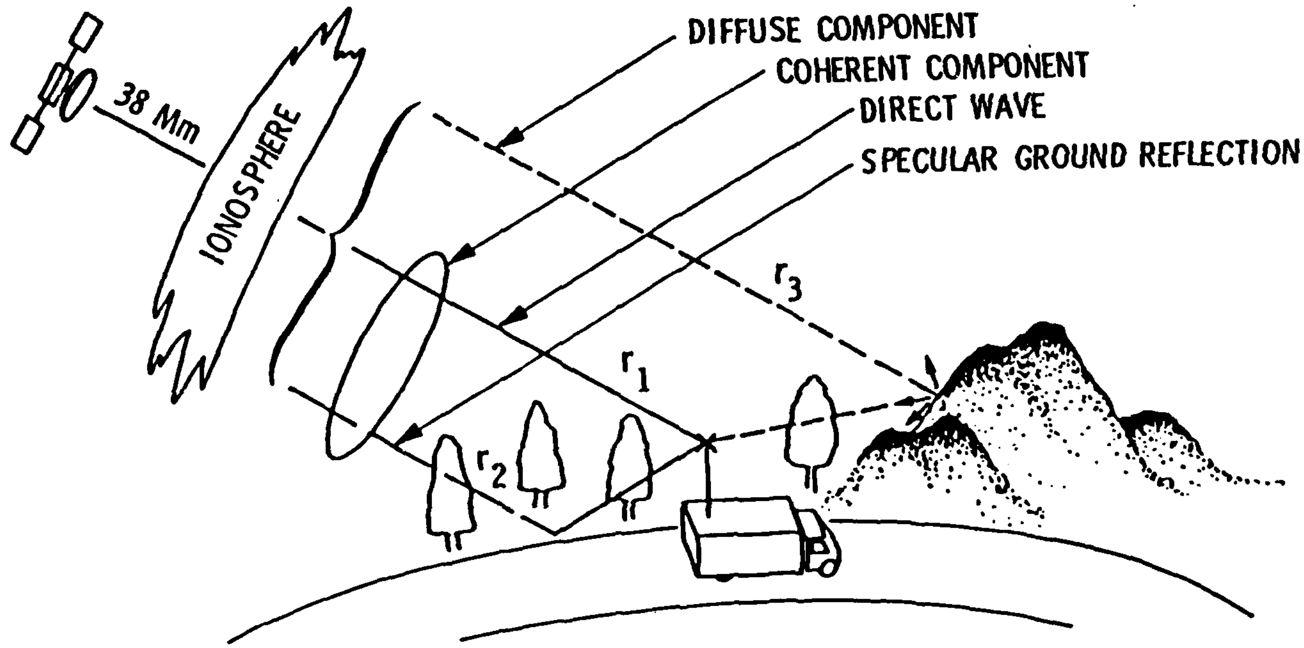
HUGHES AIRCRAFT COMPANY

FULLERTON, CA 92634

* Based on an original collaboration with Michael J. Andresen and Thanh T. Tran

JPL

PROPAGATION EFFECTS ON LAND MOBILE SATELLITE SYSTEMS



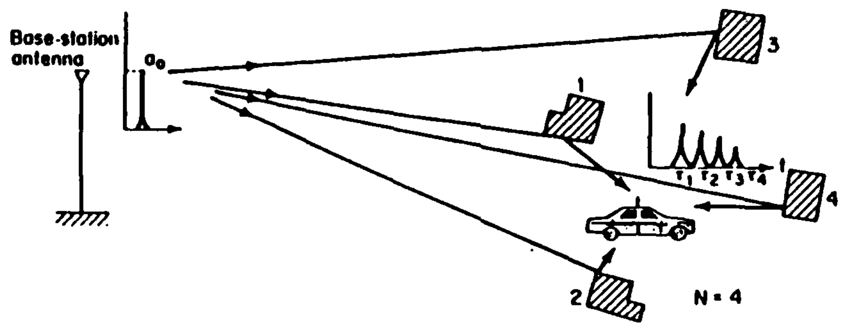
MOBILE SATELLITE INDUSTRY BRIEFING
SPONSORED BY NASA/JPL

NOVEMBER 13 - 14, 1985

Dr. Ernest K. Smith

SLIDE 2

$$s(t) = a_0 \sum_{i=1}^{n} a_i \, \delta(t - \tau_i) \cdot e^{j\omega t}$$
$$= E(t) \cdot e^{j\omega t}$$



$$d = \int_0^\infty t E(t) \, dt$$

$$\Delta^2 = \int_0^\infty t^2 E(t) \, dt - d^2$$

Parameter	Urban	Suburban
Mean delay time d	1.5–2.5 μs	0.1–2.0 μs
Corresponding path length	450–750 m	30–600 m
Maximum delay time (−30 dB)	5.0–12.0 μs	0.3–7.0 μs
Corresponding path length	1.5–3.6 km	0.9–2.1 km
Range of delay spread Δ_s	1.0–3.0 μs	0.2–2.0 μs
Mean delay spread	1.3 μs	0.5 μs
Maximum effective delay speed	3.5 μs	2.0 μs

MOBILE COMMUNICATIONS ENGINEERING *William C. Y. Lee*

SLIDE 3

PHEMOMENA

HUGHES

MULTIPATH:	FADING WITH MOTION
	SELECTIVE FADING IN FREQUENCY
	PULSE DISTORTION
	DIFFUSE DOPPLER
SHADOWING:	SLOW FLAT FADING WITH MOTION
SPECULAR COMPONENT:	LINE OF SIGHT PATH IN ADDITION TO SCATTERERS

MEAN EXCESS PROPAGATION LOSS

F. AMOROSO/GROUND SYSTEMS GROUP

SLIDE 4

CHARACTERIZATIONS

HUGHES

MULTIPATH:	PULSE POWER PROFILE VS. POSITION (OR TIME)
	COMPLEX PULSE ENVELOPE VS. POSITION (OR TIME)
	STATISTICS OF RECEIVED POWER WITH POSITION (RAYLEIGH, RICIAN)
	FREQUENCY TRANSFER FUNCTION VS. POSITION (OR TIME)
	CORRELATION BANDWIDTH
	SPECTRUM OF DIFFUSE DOPPLER WHEN UNMODULATED CARRIER IS TRANSMITTED
SHADOWING:	STATISTICS - - LOGNORMAL
	FADE RATE
SPECULAR COMPONENT:	RATIO OF SPECULAR TO MEAN SCATTERER POWER (RICIAN K FACTOR)

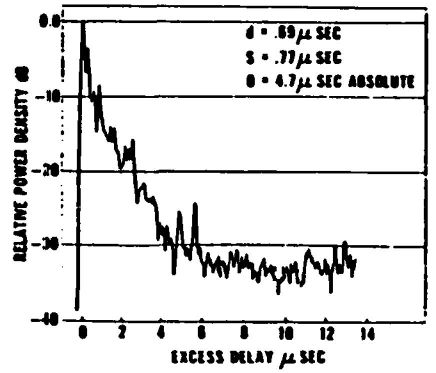
MEAN EXCESS PROPAGATION LOSS: LARGELY EMPIRICAL

SLIDE 5

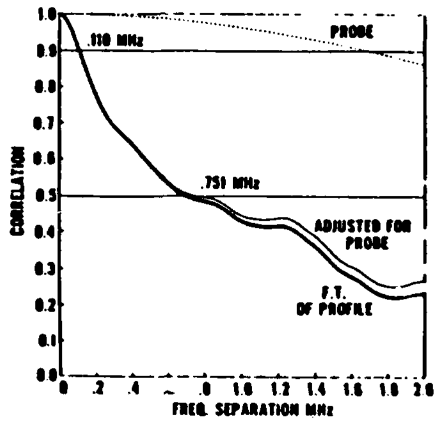
F. AMOROSO/GROUND SYSTEMS GROUP

IEEE TRANSACTIONS ON COMMUNICATIONS, NOVEMBER 1975

COX AND LECK: MULTIPATH PROPAGATION STATISTICS FOR RADIO CHANNELS



(a)



(b)

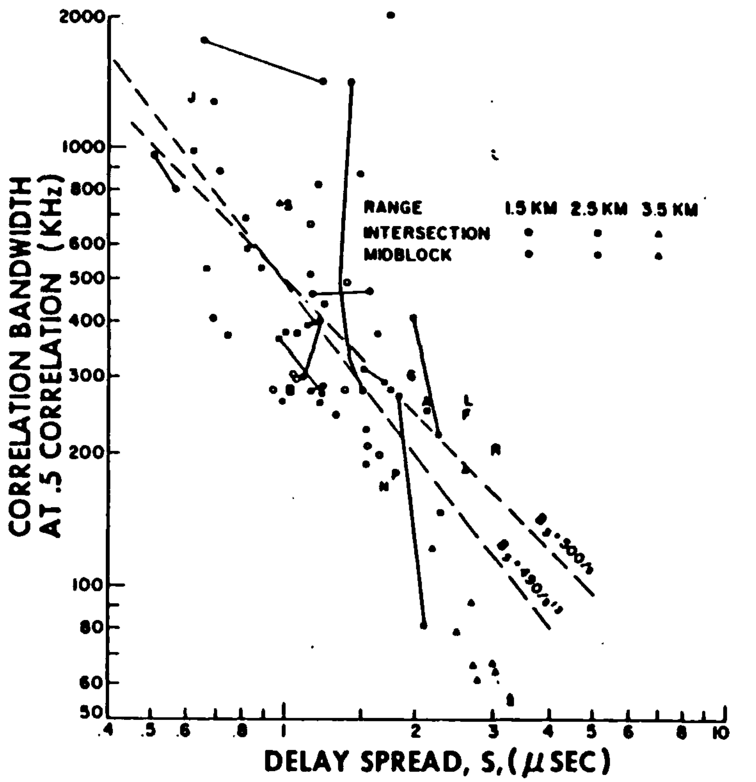


Fig. 5. Scatter plot of correlation bandwidth at 0.5 correlation versus delay spread.

SLIDE 6

[From page 123] VOGEL MICROWAVE JOURNAL • OCTOBER 1985

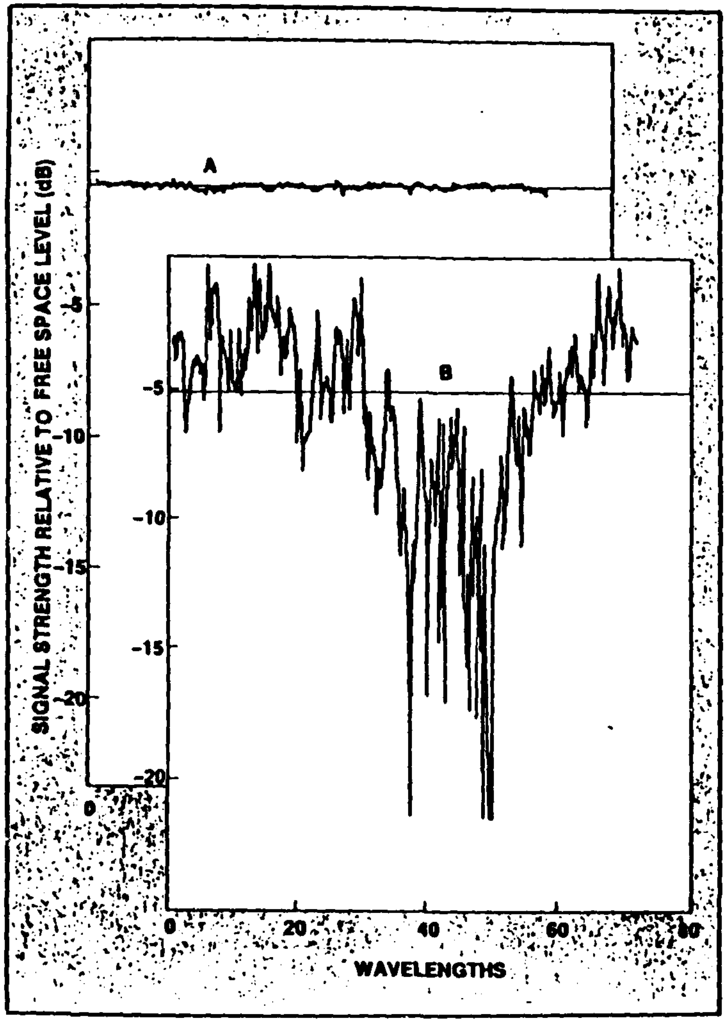


Fig. 9 One second time plots of received signal. Shown in A is a signal without blockage. B shows signal in a pine forest. Clutter is due to signal blockage.

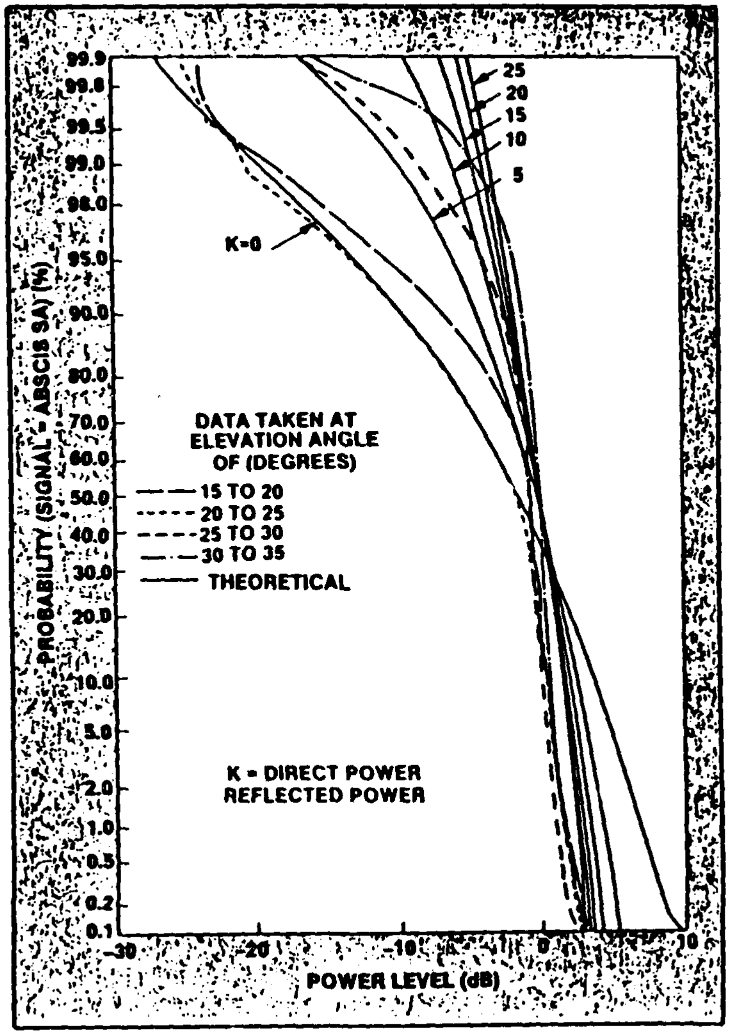
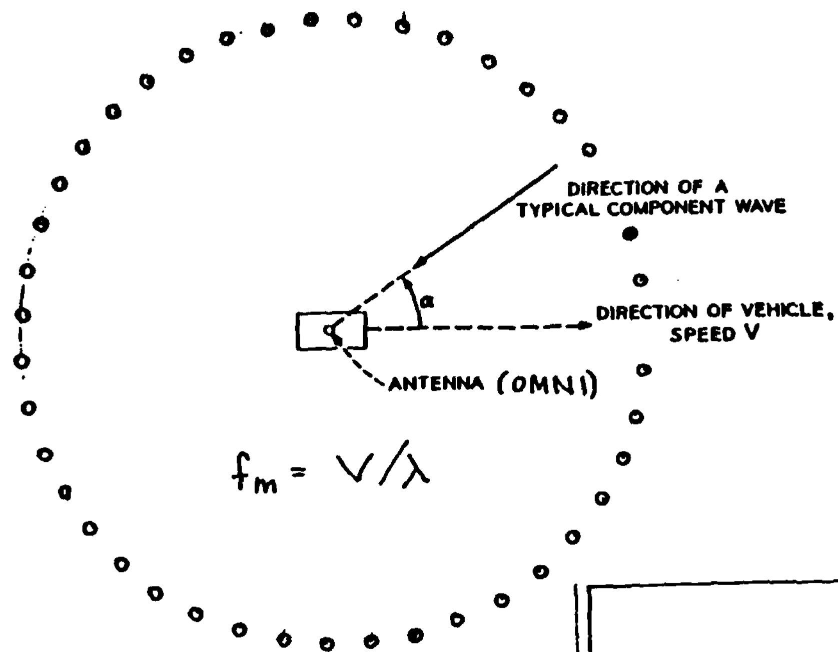
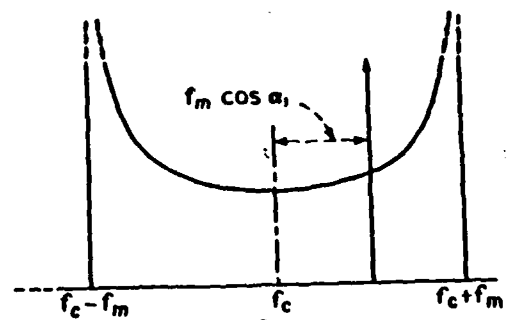


Fig. 11 Ratio of specular to random power for various K values.

SLIDE 7



$$S_o(f) = \frac{1}{\pi f_m \sqrt{1 - f^2/f_m^2}}$$



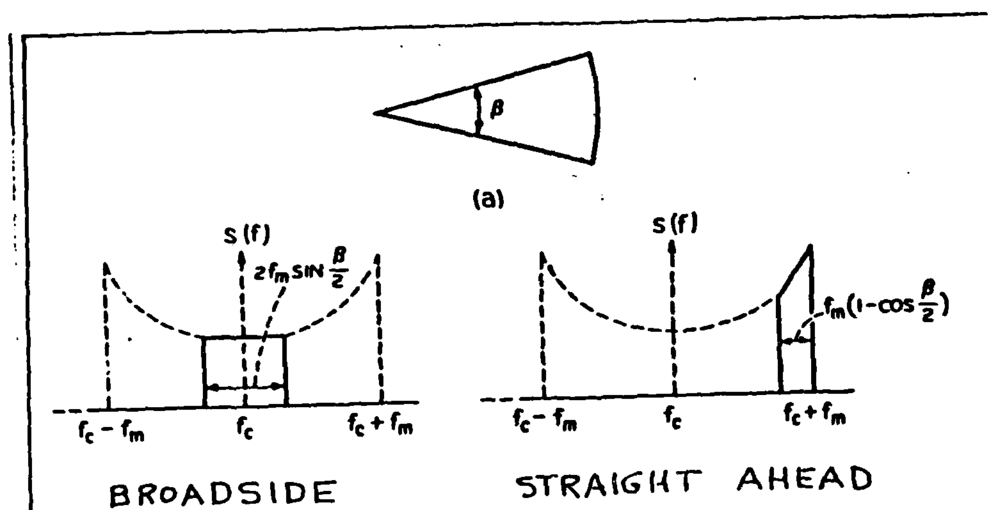
(a) INPUT SPECTRUM

A Statistical Theory of Mobile-Radio Reception

By R. H. CLARKE

THE BELL SYSTEM TECHNICAL JOURNAL,

JULY-AUGUST 1968



SLIDE 8

IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. VT-22, NO. 4, NOVEMBER 1973

A Multipath Fading Simulator for Mobile Radio

GASTÓN A. ARREDONDO, MEMBER, IEEE, WILLIAM H. CHRISS, MEMBER, IEEE, AND EDWARD H. WALKER, MEMBER, IEEE

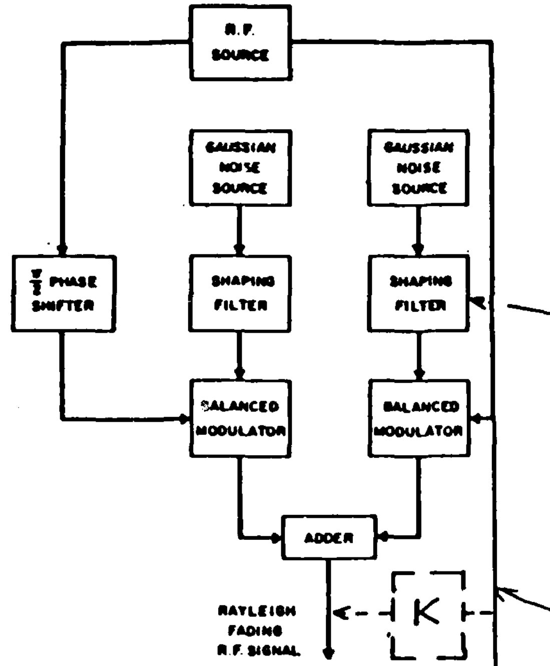


Fig. 1. Fading simulator block diagram.

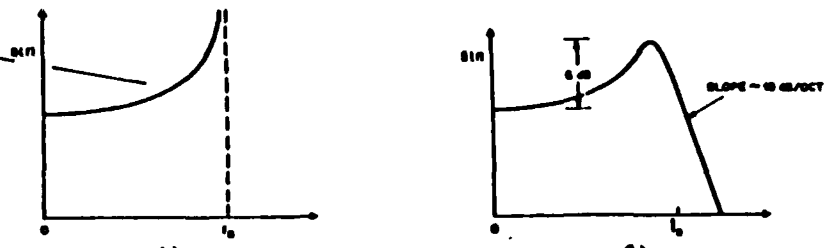
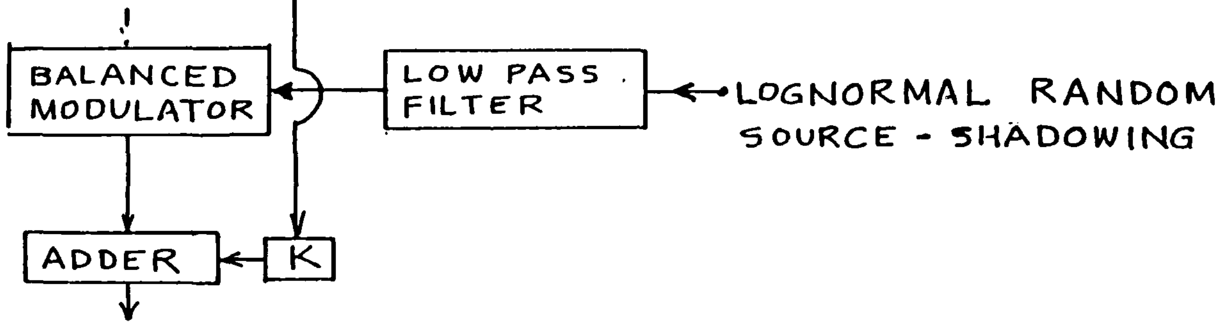


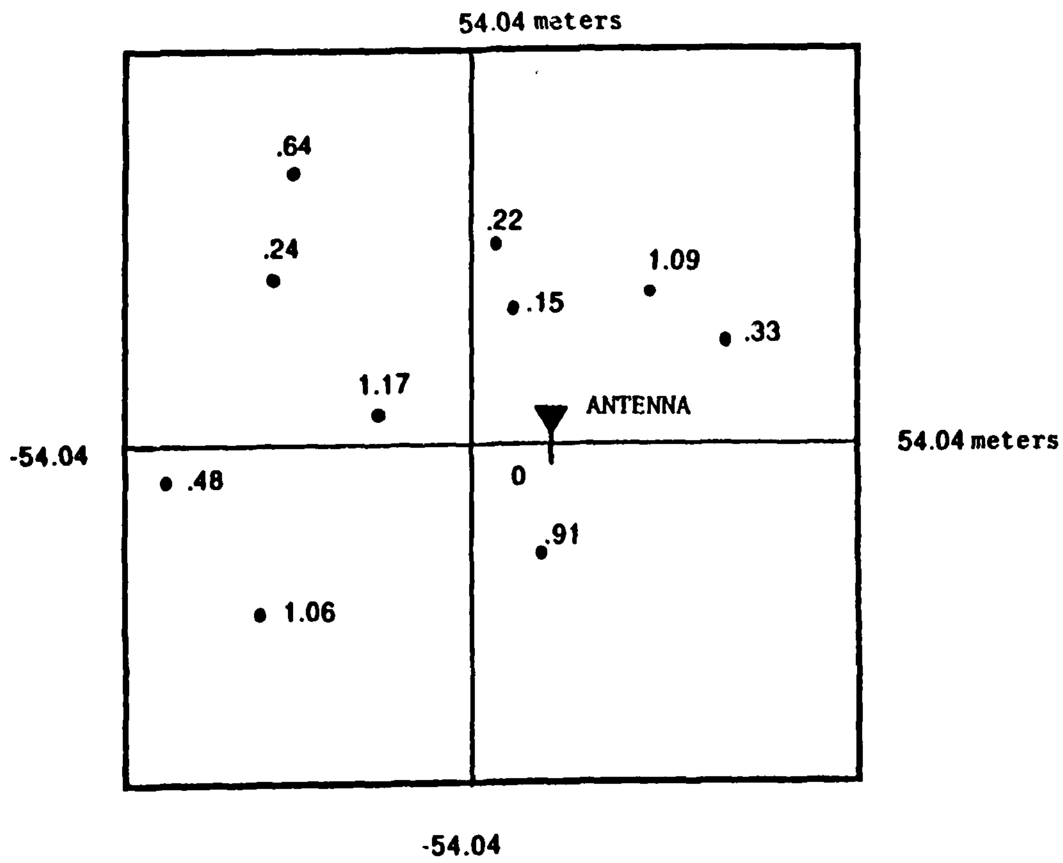
Fig. 2. Theoretical and simulated spectral densities. (a) Spectral density of complex envelope. (b) Shaping filter frequency response.

FREQUENCY SHIFT ?



FADING, RAYLEIGH, RICIAN, SHADOWING, DOPPLER

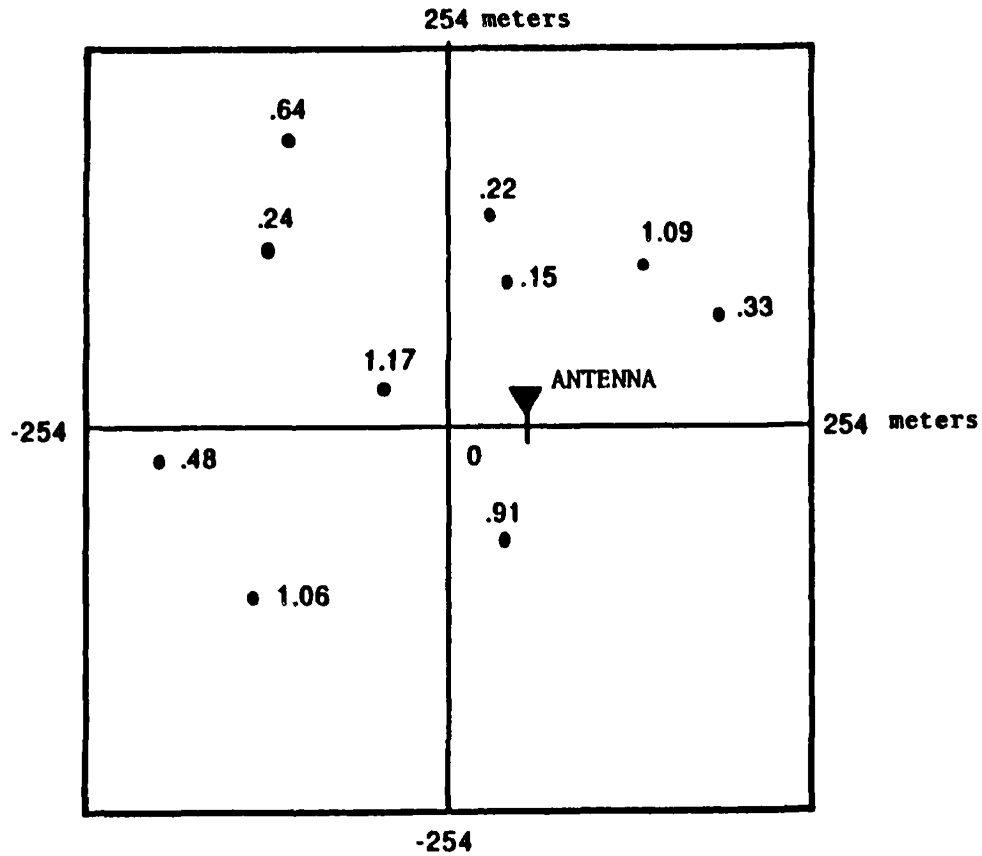
SLIDE 9



ANTENNA POSITION = (13.69, 1.80)

Figure 1. Constellation 1, delay spread = 40 ns.

SLIDE 10



ANTENNA POSITION = (71.72, 9.43)

Figure 2. Constellation 2, delay spread = 200 ns.

SLIDE 11

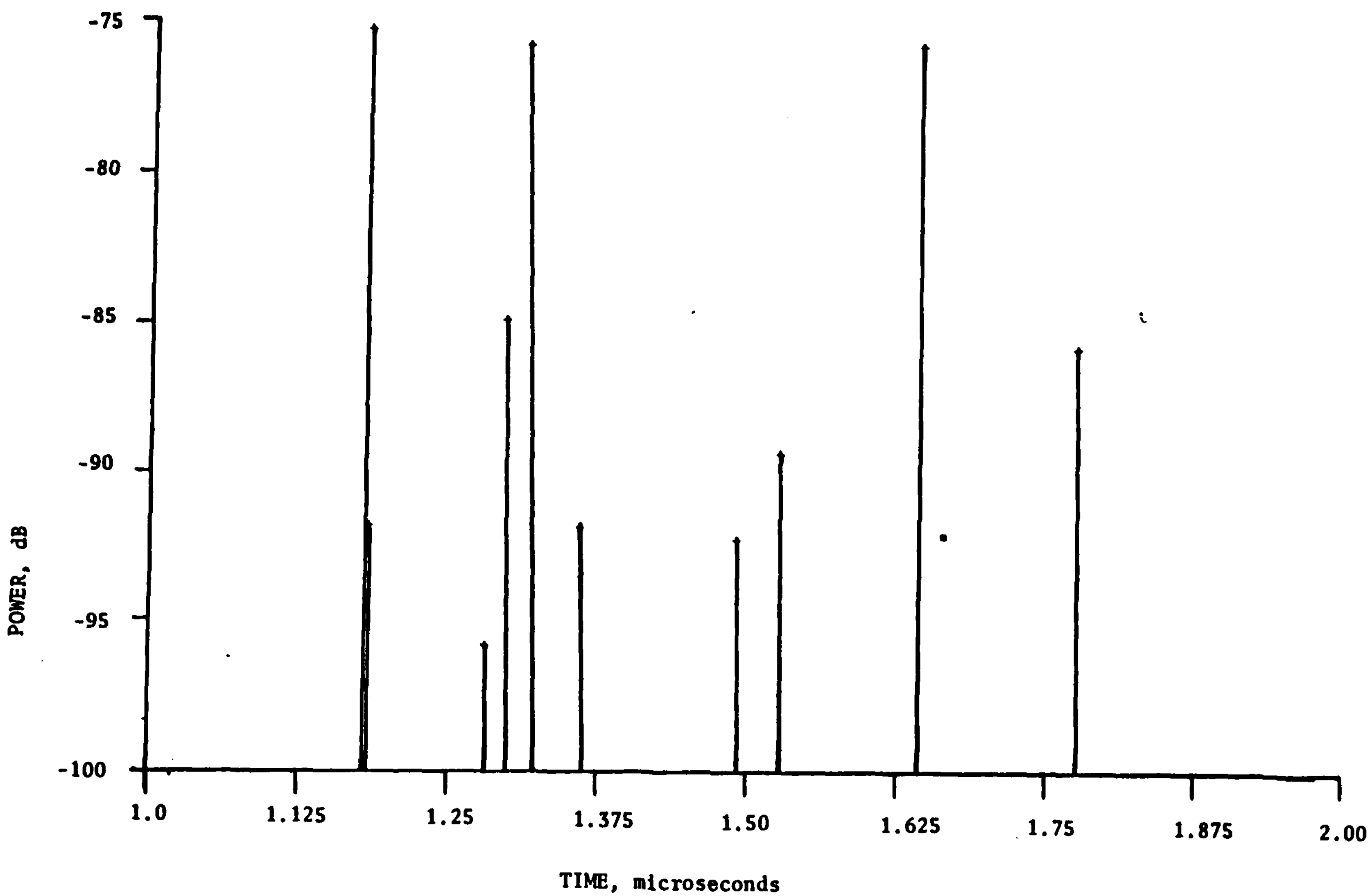


Figure 3. Pulse power envelope, Constellation 2.

SLIDE 12

$$E(t)=\sum_{j=1}^{10} B_j \ \delta(t-\tau_j)\exp j[(\phi_j-2\pi\tau_j f_c]$$

where $B_j=A_j/[4\pi R_j/\lambda]$

ϕ_j = static phase shift of the jth scatterer
A_j = voltage coefficient of the jth scatterer
R_j = distance from the jth scatterer to the antenna
$\tau_j = \dfrac{x_j + R_j}{c}$
x_j = horizontal distance from the left boundary of the constellation to the jth scatterer
f_c = radio frequency carrier, 1547 MHz
c = speed of light
λ = radio wavelength
$\delta(x)$ is a unit impulse function

SLIDE 13

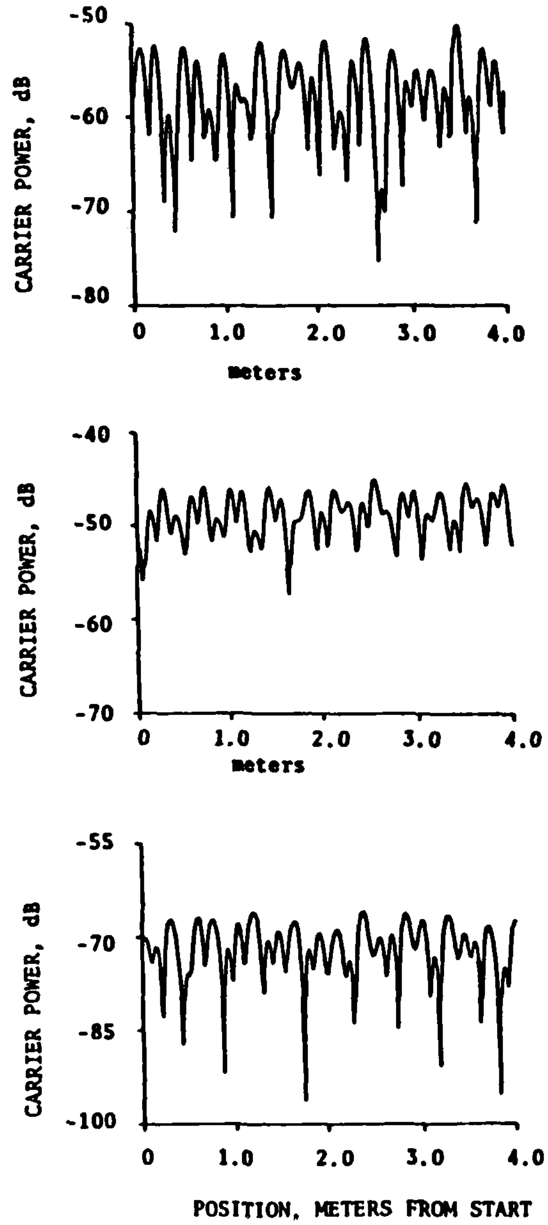
$f_c = 1547$ MHz

(a) Constellation 1, Rayleigh,

(b) Constellation 1, Rician K = 5,

(c) Constellation 2, Rayleigh.

Signal strength vs. antenna position



SLIDE 14

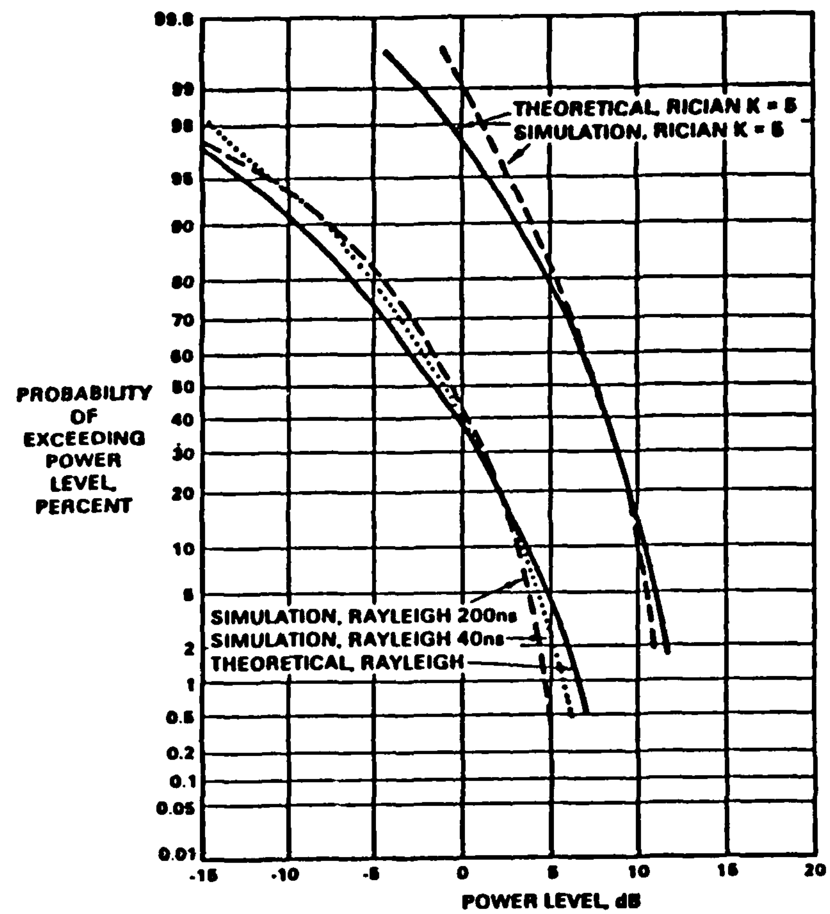


Figure 5. Validation of the scattering model by fading statistics.

SLIDE 15

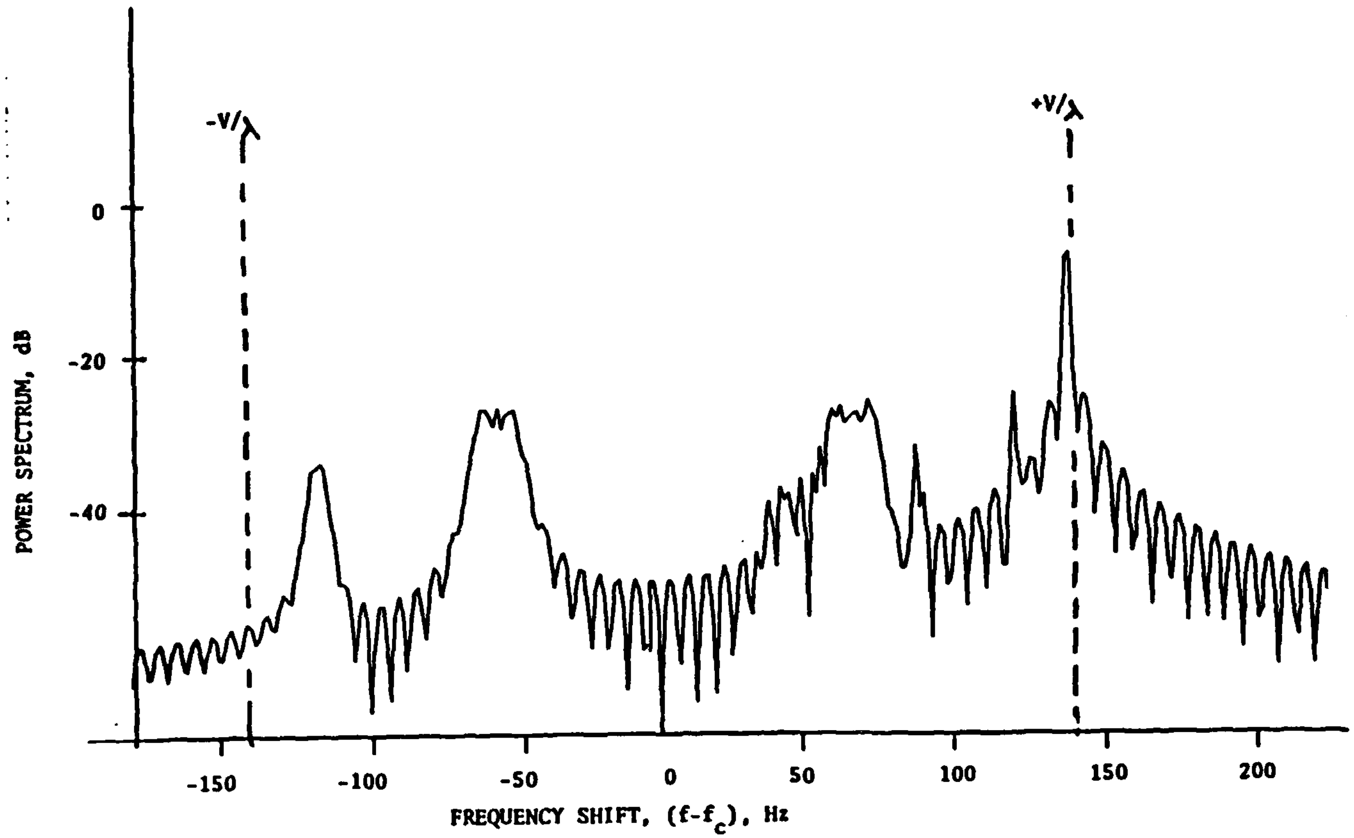


Figure 6. Received power spectrum, Constellation 2, 60 mi/h, specular Doppler = 138 Hz, unmodulated carrier transmitted, Rician K = 5.

SLIDE 16

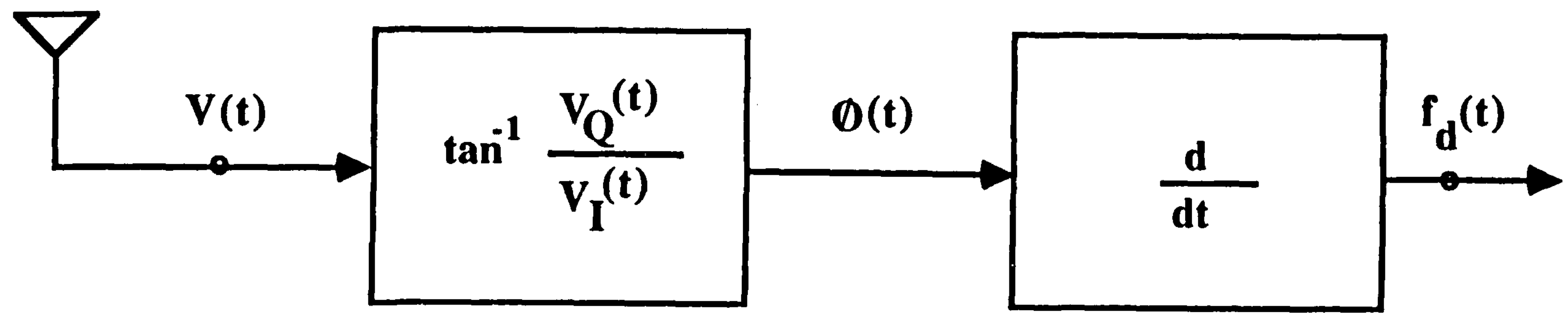


Figure 7. Extraction of instantaneous frequency.

SLIDE 17

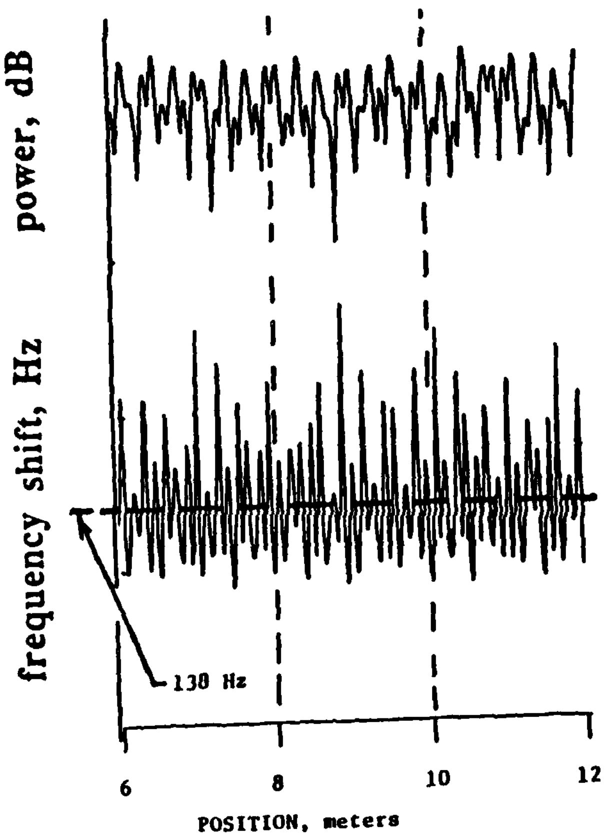


Figure 8. Instantaneous power and instantaneous frequency, Constellation 2, 60 mi/h, Rician K = 10.

SLIDE 18

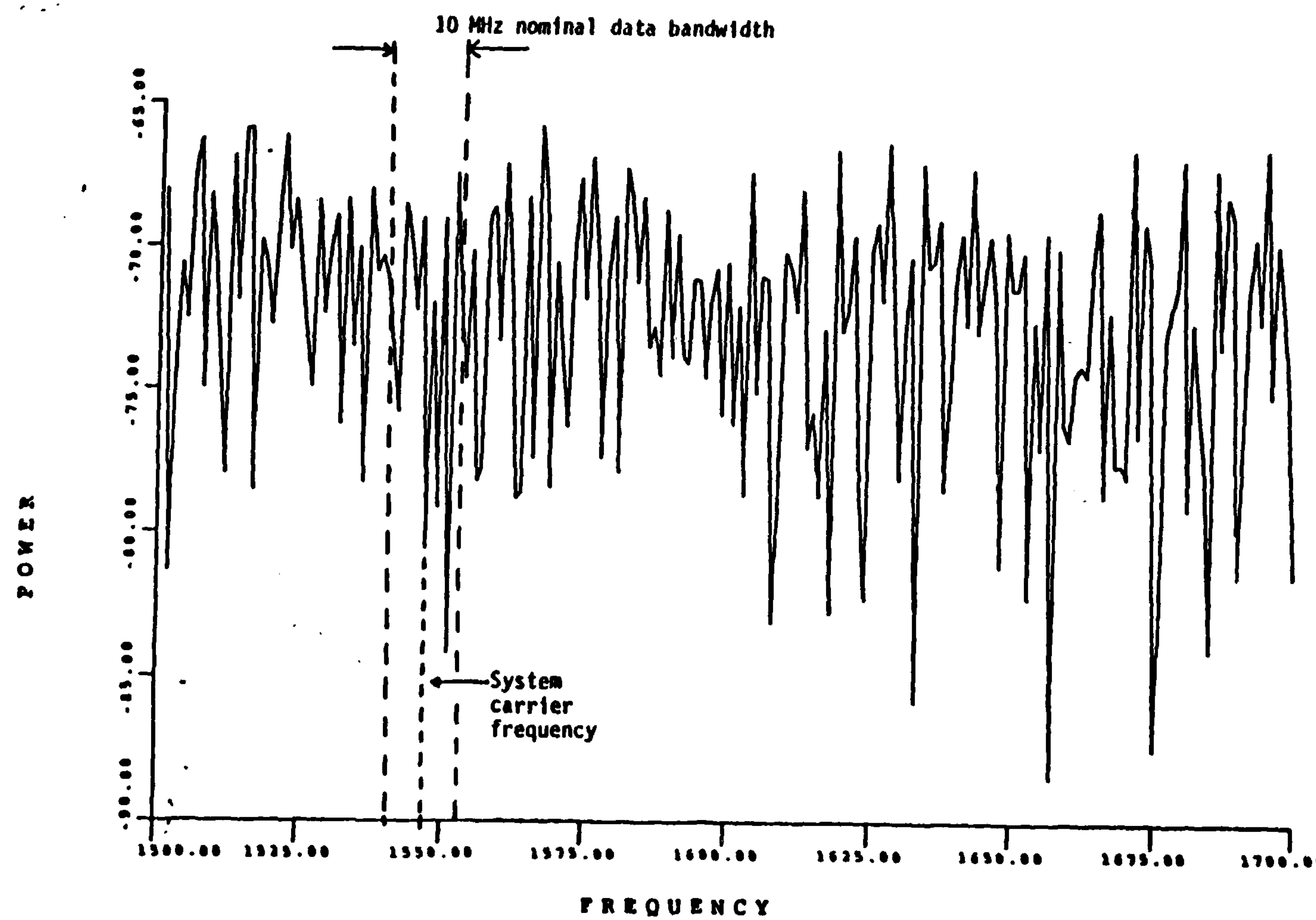


Figure 6. Swept propagation path frequency response to the antenna port, Constellation 2, antenna position as in *Figure 2*.

SLIDE 19

(a) Unmodulated Carrier

(b) 10 Mb/s
QBL Modulation

(c) 10 Mb/s
QBL Modulation

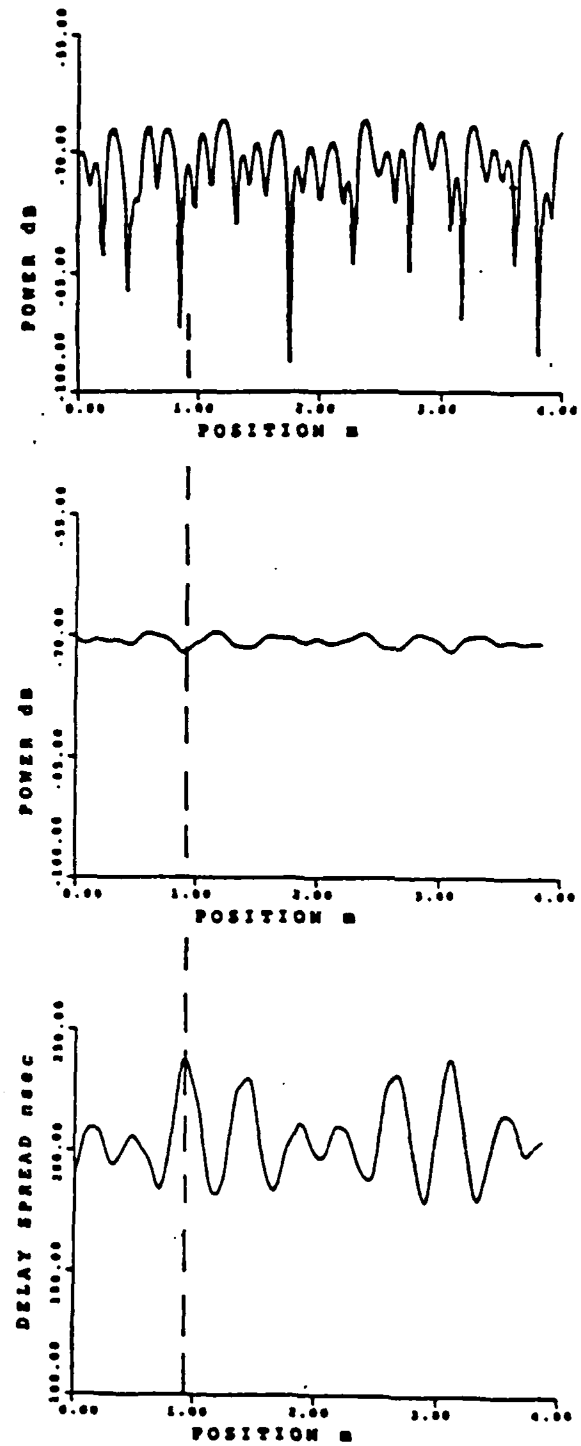
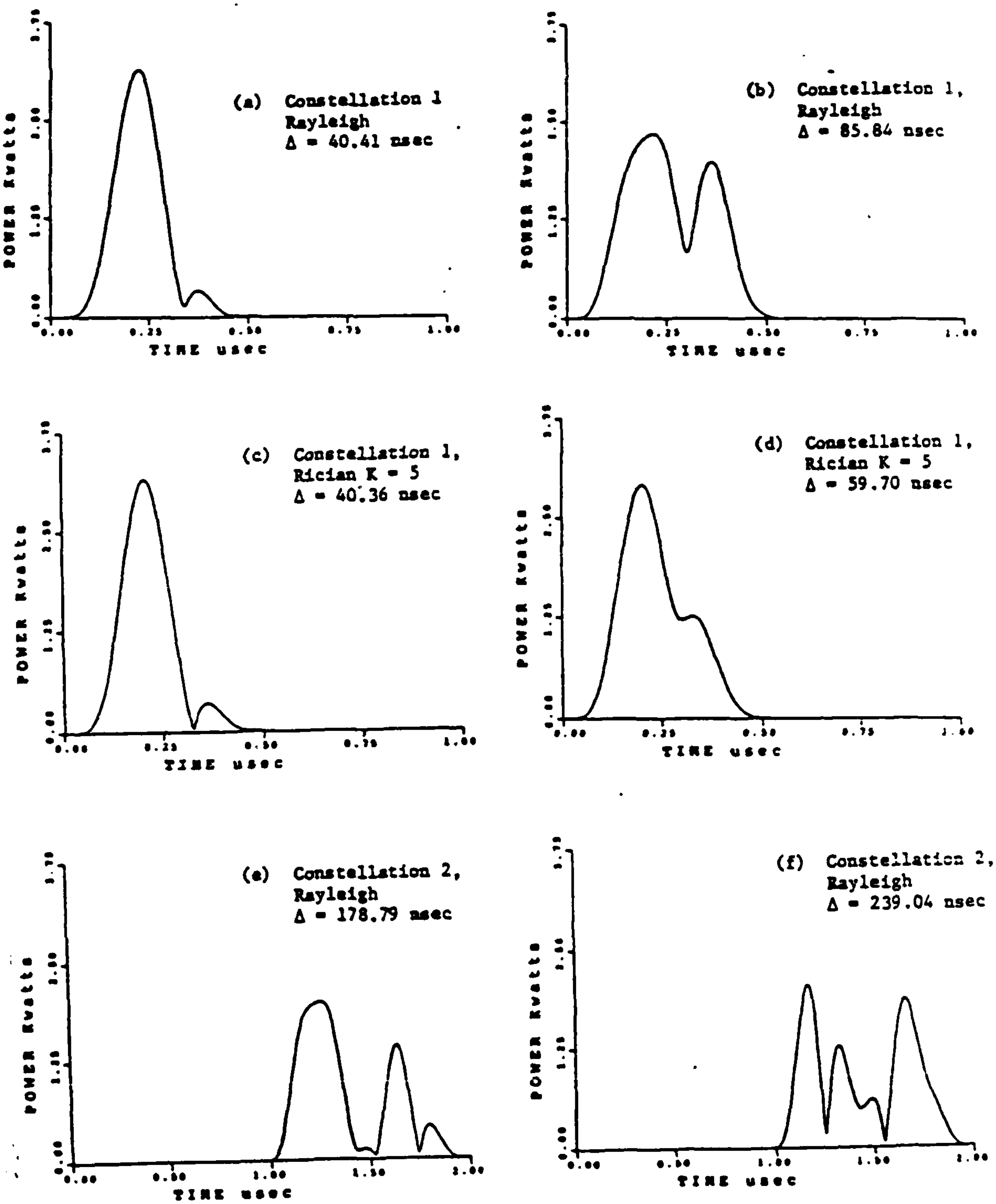


Figure 10. Received power and delay spread for unmodulated and modulated cases. Constellation 2, Rayleigh environment.

<u>SLIDE 20</u>

Figure 11. Received QBL Pulse Envelopes.



SLIDE 21

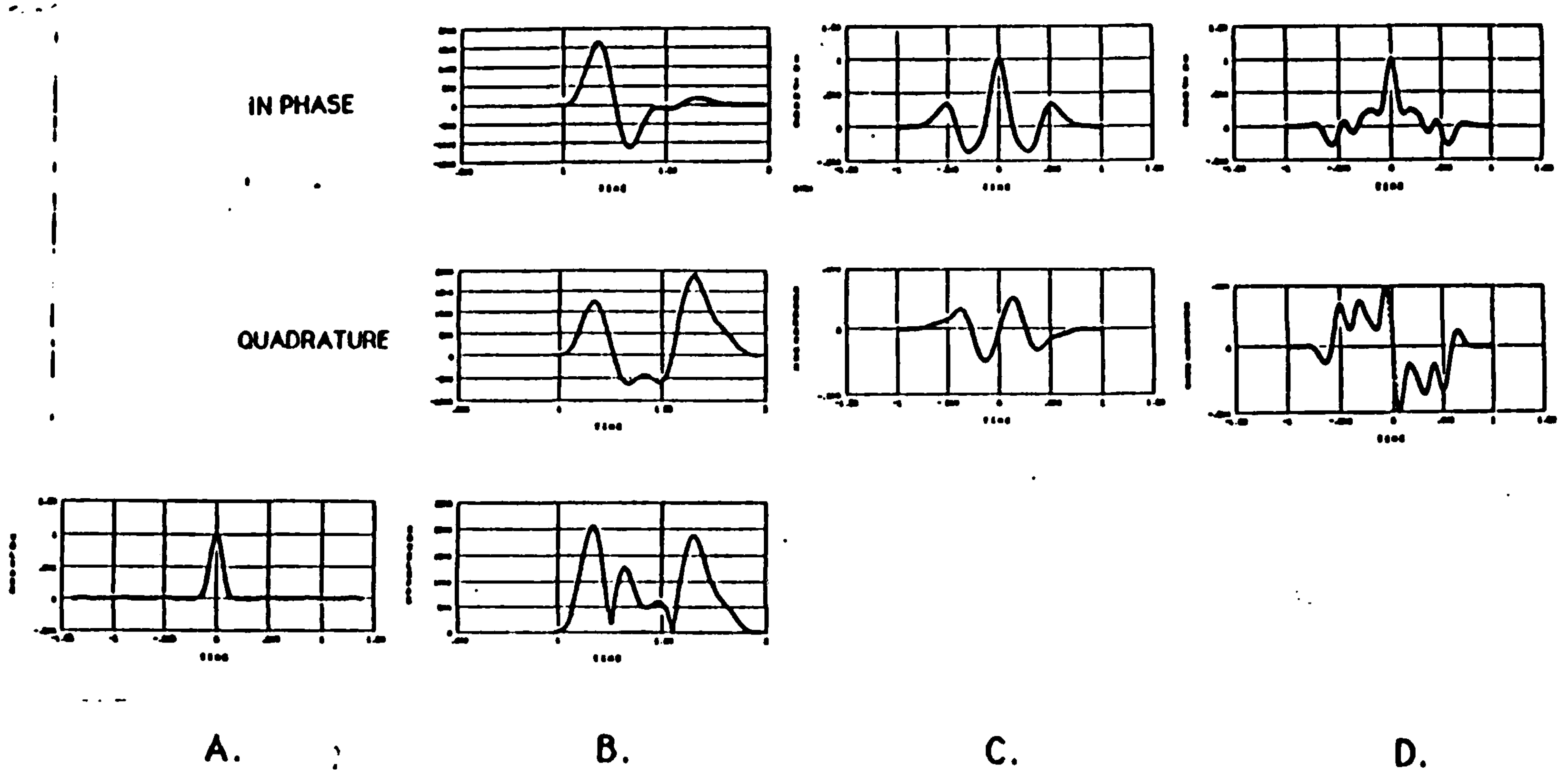


Fig 2. Signaling waveforms. (a) Transmitted QBL pulse, (b) Waveform received via 200 ns dispersive medium, (c) Matched filter output, (d) Rotating phase demodulator output.

SLIDE 22

DLE] of pseudonoise stream and we find out that the arriving power is now very, very constant. The fading is gone as far as measuring the power is concerned. We are also able to show that with this antenna path over 4 meters, the delay spread as a function of position is quite variable and in fact, the delay spread tends to be maximum when the power is minimum. This means that the pulse is being distorted, and the specific shape of distorted pulses is changing very drastically as the vehicle moves along. So this particular model would get very deeply into pulse shapes.

Here are some sampled pulse shapes [SLIDE 21]. They change a lot if the vehicle moves. These for example are pulsed power envelopes from two different places on the same track, with constellation 2. The delay spread here [LOWER LEFT] was 178 nanoseconds, and here [LOWER RIGHT] it was 239, and the pulses are shaped like differently. But our model didn't have shadowing in it, it didn't have directional antennas in it, it didn't have some of the other things that I mentioned, and so there you are... If you want to do modeling, you have to think very carefully about what it is you are trying to simulate and what effect you are trying to get at. That's all I want to say.

SHANMUGAN: That's an excellent example of the level of detail that you have to include in modeling. A lot of times people like myself and a lot of you in the audience who are involved with typical satellite design at C and K-band frequencies tend to ignore what Mother Nature has for us by way of channels. Unfortunately at lower frequencies and high frequencies the introduced by the channel phenomena have as much impact on your system performance as any functional block that you might build and put into the system. So that was a very good presentation which also pointed out the level of detail that we have to put into databases if we are going to have databases containing channel models. Questions?

MOHANTY: Two parameters, one the duration of the signal, and another the decorrelation time. And another thing, I don't know if I agree with you, you said that the bandwidth of the signal is larger than the decorrelation frequency bandwidth. I think that's where the most severe fading or distortion will occur. So you should have it the other way, the decorrelation bandwidth should be larger than the bandwidth of the signal.

AMOROSO: Point well taken. If all you're measuring is received power and you have a very high rate data stream, a data stream whose bandwidth is much wider than the correlation bandwidth, then you'll see that power quite constant as the vehicle moves along. However, the individual pulses can be very badly distorted, and in fact the shape of the pulse can change drastically within a wavelength of motion, a wavelength of displacement. So you might find yourself having to work pretty hard to get the data back out of the distorted pulses. Now we were concerned here with what was direct sequence pseudonoise, where one bit may have consisted of thousands of pseudonoise chips. The idea was to determine the value of a *data bit* that was in fact quite long compared with the delay spreads. That obviously is a very different problem than receiving data transmitted at the assumed *chip rate*, i.e., one chip per bit. I agree that distortion is very severe if the data rate is greater than the correlation bandwidth, but the received power level alone, measured with vehicle displacement, will be much more constant. In a sense that proves the point that the

model you use and the result you are seeking have an awful lot to do with the communication system. What is the objective of the communication, what exactly are the parameters? I feel that's quite a tacky area. That was a long-winded answer. I hope that's a good question, because it's the only one I have time for!

SHANMUGAN: Our next speaker is Phil Balaban. Phil is going to talk about future directions in CAD tools for transmission systems engineering. In particular he's going to talk about expert systems and their role in CAD tools.

BALABAN: When Sam called me up to tell me that he wanted to put me on the panel he asked me, "What do you think?" I thought of his new system BOSS, with which I was very impressed because we started out in a very complicated beautiful system called SYSTID which was very hard to use (BOSS took most of the things out of it, is much easier to use than the Batch oriented SYSTID system) and he asked me, "What do you think the new BOSS is lacking?" I said, I think it should have an expert system applied to it. He then told me to talk about this even though as I explained to him I didn't know anything about it, it was just an idea of mine. Since I am not an expert in expert systems I will talk from the user's point of view and I will only say a few words.

All CAD systems, including BOSS are designed to be used by gurus. A communication system engineer without knowledge of the CAD system cannot use it. There are many, many things embedded which are particular to the specific system. I think that a simpler tool should be used. Somebody mentioned a very nice word, fast prototyping done by a junior engineer who has no time to learn all the peculiarities of the system. He should know only what the communication system is, everything else should be taken out of it. This could done either automatically or through a conversation. I went to BELL LAB to ask what an expert system is, since there are many people who work on it. I talk to many experts and everybody gave me a different point of view, so there are as many definitions of the expert system as there are experts. Is this true? Actually in the seventies at BELL LABS an expert system was used, without knowing what it is, for line maintenance. The maintenance operation was performed through conversation. The procedure started an the terminal and led to the diagnosis of what's wrong. Right now, it is applied mainly actually to networks and we have an expert who will talk about networks, network management. Our first speaker from PACIFIC Bell talked about it. (note GE ? NBP) That's one of the major applications is network systems, network management. By the way there is also something in the chip design. Chip design is ruled by experts. But they burn out and don't recommend their expertise to anyone else. This should be embedded in an expert system.

There is something else also operated by gurus of the expert systems. Usually the people who come to gurus are also experts in communication systems. So an acknowledged communication engineer is left out completely. I want to talk about how a system like this can be improved. I don't know if this belongs to this session or to the earlier one. It refers to the near future. With the way things are working now, if somebody wants to use it he comes to the University of Kansas to the guru, and they say "So you want to use BOSS?" [laughter] I divided the expert system into three phases.

The first one is immediate, just making the system easier to use. What I mean by that is that all too specific knowledge from a system should be eliminated either automatically, (it can be done). If this cannot be done automatically, experts tell me that's not an expert system. That's a problem since it can only be done through conversation with the expert system.

The second phase comes after the initial design of the system, on a simulation system as Mike described, when it doesn't work right. How can we optimize the system and make it better? Sometimes you know but sometimes you follow the wrong way. In a way the expert system should partially analyze and suggest where the pitfalls in the communication system are. And this is something which some of the TRW people mentioned a great deal. I think what is missing from what we have at Bell Labs and from BOSS is probably memory for all these beautiful models. We designed systems together with all these subsystems and components, and there is no memory there, no storage. If somebody designs a subsystem, a filter modulator/demodulator which can be useful, this should be in the database, in a form easy to use and the system should search for it, and find it for him (the user). There was a good question in the previous session, that the system design has a lot of wheat and chaff. We did something similar when I worked with some CD systems for LSI design: for every system, we defined two types of models. One model was a detailed model for the system, and the other one was functional, for the same system. The functional model should be used for this type of system. So whatever system is designed, we should define the model and put it in. And essentially it is sort of a dream. When a new engineer comes in and wants to have all the information, but has no information that he wants, he should propose what kind of a system can be available. A visual system radio should say, well, he can use people who have designed 64 QAM systems and so on, here is an example, I'm calling the specs. I am trying to give examples of the three phases. This is phase 1. This is a satellite system like those which Mike is using at GE, has a QPSK modulator, a filter which consist from two filters in a way, a TWT, another filter, an electric filter chebyshev, demodulator and bit-error estimator. In such a system, there are many things which have to be defined poorly, since they are knowledge-defined. First of all, a simple thing like defining a δT sampling interval is not part of the knowledge. The system should be capable of guiding the engineer. It can say for example, well your largest bandwidth is so many Hz, I picked a δT, four times larger, double the micro state. You want to have more than that, or it's good enough or something like that. It doesn't have to specify δT. We have in system like BOSS 4, 5, 6 different bit-error estimators, fully analytic, semi-analytic, important sampling you name it and they are all used for different things. The user doesn't have to know what kind of estimator is used for that. You can ask me some questions like, is the system linear? Is the noise Gaussian? All right, you can use the semi-analytic, if it's not using another estimator. But if it is for important sampling, where do you think your errors come from? For things like that conversation with the system is used to find the errors. In this I think power and scaling has to be used to get over the estimator. What kind of signals do you want at the input? Usually we use PN sequences or something like that depending on how much intersymbol interference exists to guide him out to

define the PN sequence. The system in reality doesn't use PN sequences at the input. Filters: we have a presentation on frequency domain filters, FFT filters, time-domain filters. It's not a title, he knows the title. What kind of presentation do we want in simulation? Do you have overlap and save, overlap and add, all kinds of things? Again the system should guide you as to what to use.

This is what I call the expert system approach. What we use is the knowledge in the response to pattern. If he wants to have a specific subsystem, then the pattern is what he looks for having a subsystem like that. In a way, the conversation is very simple with that. The expert system is being converted to someone else's rules.

Now this [VISUAL AID] is what I call retrospective design. You design the system and want to improve it.

This is phase two. The system should be able to perform sensitive analysis and then optimize the performance according to the specificity and then search the database for similar designs mentioned before. And this is something that was mentioned in the first session. I suggest, stronger, cheaper cost-effective subsystems and components which were designed before. Maybe as an example, the sensitivity analysis show that error-rate is dominated by adjacent channel. That's how it is done. It can optimize the outer filter to reduce adjacent channel interference causing interference to go up so it can develop optimization to balance that.

This is phase three. A system should guide the designer, or give a specification to find the design. And it should also show, depending on whether he is a novice for this system or not, the database. If not this can be a pitfall to the system and should be investigated. I have an example here. Since I work in digital RF now, we are given a digital radio, and fast frequency bit-rate. we can suggest the use of similar systems, and we have to investigate the fading interference reflections for the transmitter/receiver, the intersymbol interference, things like that. The system, if the user is novice, should make him aware that these are the systems. So the knowledge that a system should have at different levels of access, and somebody who is a guru doesn't need all these things he says, I'm a guru don't give me advice, I know what I am doing. But for someone who is not (novice) the system can have different levels where he can go in and stop at each level. And besides this is sort of a tree of decisions and I think the diagnostics of the system should be made available on demand. I made the decision for you because I've made these rows. So you can agree with them or you can backtrack, and say you don't like this decision, give me something else. The we'll go again to something else, and you can backtrack at any point you want. So if a system like that is built, and I think it will be built by an expert laboratory. This is from the New Yorker the expert's research laboratory there where they have special systems. The caption was "don't mind me folks, I am here just to read the meters".

What do you want to do is to have a conversation. You wouldn't feel like you're in a prison any more because you'll talk to the system. My advise is to tailor it to the recipient, based on knowledge which is in the database. And realistically, I think you have higher productivity and low-cost information because in a way the novice or any designer will work on a lower level than a system is designed for. So the impact on users is that the user who will see this sys-

↓ cut ↓

5/22/87
AT&T
FOIL 1

Future Trends in Computer - Aided Modeling for Communication Systems

Consultation (Expert) Systems

Philip Balaban
AT&T Bell Laboratories
Holmdel, NJ 07733

P. Balaban 5/27/87

5/22/87
AT&T
FOIL 2

PROBLEM

- Ideally CAD tools have to be used by communication systems engineers
- CAD tools designed to be operated by CAD experts (gurus)
- The interface to the CAD expert is usually a communications system expert.
- No help provided to novice inexperienced engineers

P. Balaban 5/27/87

SLIDE 2

5/22/87
AT&T
FOIL 3

FUTURE CAD TOOLS
(Consultation System)

- Phase 1
  - Only communication system design knowledge should be required to use the system.
  - All tool specific knowledge should be derivable from interactive system description
- Phase 2
  - Guide the user to a better (optimized) design.
  - Help select standard or suggested systems and components
- Phase 3
  - Guide a novice Communication Engineer to select a system.
  - *Suggest what areas of the system require special attention.*

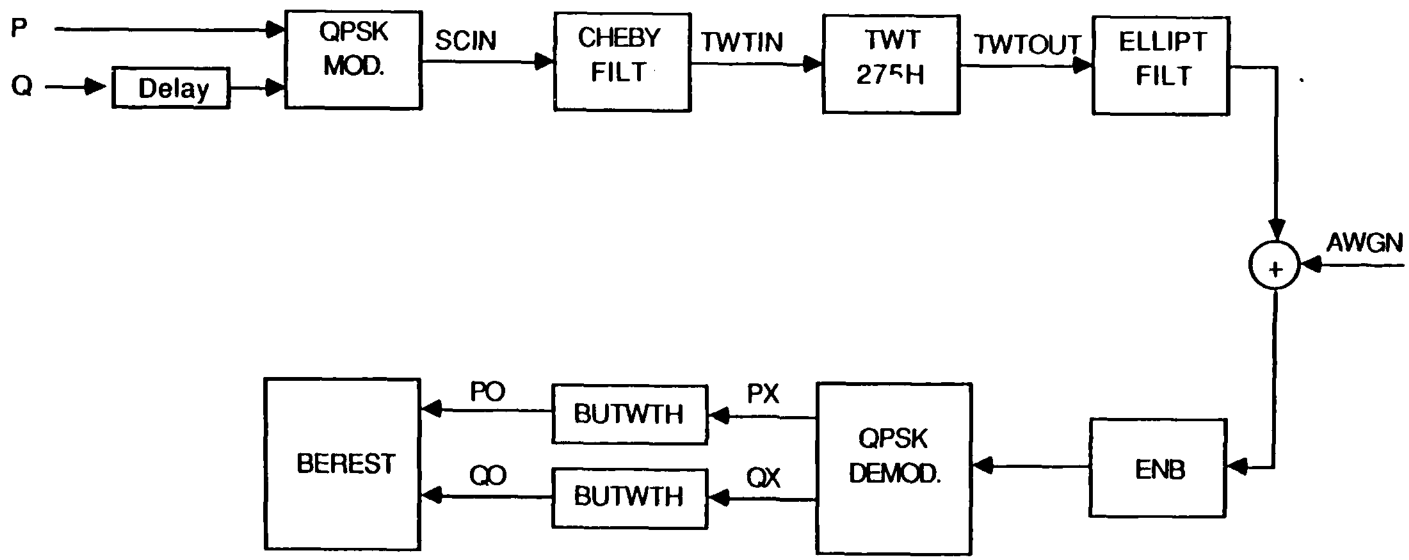
P. Balaban 5/27/87

SLIDE 3

Phase I

Example

Satellite System Block Diagram



SLIDE 4

Knowledge Specific to Tool

- Selection of sampling interval
- Selection of a bit-error-estimator and its parameters
- Noise bandwidth estimator
- Signal power estimator
- Signal level scaling
- Type and duration of input signal
- Frequency or time-domain filters
- Simulation duration

P. Balaban 5/27/87

SLIDE 5

EXPERT SYSTEM APPROACH

- Knowledge in problem solving
  - Knowledge of responses to patterns
  - Knowledge of which patterns to respond to first
- Advantages:
  - Fairly independent rules
  - Quick solutions to complex problems
- Key idea:
  - For carefully chosen domains, the uniform interaction of many simple "if...then..." rules can replicate expert performance!

SLIDE 6

Consultation Systems
Phase 2

(Retrospective Design)

System should be able to:

- Perform sensitivity analysis
- Optimize performance
- Search through database for similar designs
- Suggest standard (cost effective) subsystems and components from database

P. Balaban 5/27/87

SLIDE 7

5/22/87
AT&T
FOIL 7

EXAMPLE

- Sensitivity analysis shows error-rate dominated by adjacent channel interference
- Optimize RF filter to balance contributions from adjacent-channel interference and inter-symbol interference
- Suggest to use an RF filter from database

P. Balaban 5/27/87

SLIDE 8

5/22/87
AT&T
FOIL 8

CONSULTATION SYSTEMS
Phase 3
Prospective Design

- Guide to possible designs for given specifications and constraints
- What problems should be investigated in the selected system

P. Balaban 5/27/87

SLIDE 9

EXAMPLE

Radio Route Engineering

- Given: Digital radio, path, frequency, bit-rate etc.
- Suggest: Previously used similar systems from data base, new developments in the field, a possible design scheme and parameters
- Suggest problems to investigate (trouble spots):
  - Channel:
    - Fading, interference, reflections
  - Transmitter Receiver:
    - Problems: ISI, noise, phase jitter, etc.
    - Solutions: Adaptive equalizers, clock recovery, etc.

SLIDE 10

Knowledge Systems Should Have

- Multiple knowledge level access
- Decisions are made automatically or through interactive conversation
- Diagnosis of decision on request

P. Balaban 5/27/87

SLIDE 11

THE KNOWLEDGE LEVEL

CAN OPERATE WITHIN A LEVEL

CAN REDUCE A LEVEL TO ONE BELOW

- BUT DOING SO DOES NOT INVALIDATE THE UTILITY OF CONCEPTUALIZATION AT THE HIGHER LEVEL.

1.5

SLIDE 12

5/22/87
AT&T
FOIL 11

GOALS?

Ideal Goals

- Conversation
- Advice tailored to recipient
- Based on huge knowledge

Realistic Goals (but unlimited)

- Higher productivity
- Lower cost information

P. Balaban 5/27/87

SLIDE 13

IMPACTS ON USERS

WILL SEE SYSTEMS BECOME FRIENDLIER

- WILL NOT PERCEIVE A RADICAL DIFFERENCE

USERS WILL BECOME MORE NUMEROUS

USERS WILL HAVE MOST OF PROBLEMS HANDLED BY SOFTWARE

1.8

SLIDE 14

5/22/87
AT&T
FOIL 12

Impact on Consulting Engineers

- Will advise users of and on software
- Will not see routine problems
- Will see unusual problems

P. Balaban 5/27/87

SLIDE 15

5/22/87
AT&T
FOIL 13

Impact on Research Engineers

Will be able to direct research to:

- New design technologies
- Formalization of less common design techniques
- Formalization of less common problems in common design techniques

Research embodied into consulting systems

SLIDE 17

P. Balaban 5/27/87

5/22/87
AT&T
FOIL 14

Limits in the Foreseeable Future

Consultation Systems

- Will have limited knowledge
- No means of autonomous learning

SLIDE 18

P. Balaban 5/27/87

tem, will become friendlier and he will not perceive a radical difference. And I hope that users of systems like BOSS will be much more numerous and most of their problems will be handled by the software, by experts. The consulting engineer will advise users on the software which they have and will not see it as action for routine problems. If the problem is not routine and is not there (in the system), it will come to him, so he will hopefully see all the unusual problems. The impact on research is that he can create new design technologies because what an expert system is, is a design technology which was somehow intuitively defined in the system itself. So you are able to formalize less common design techniques in your system. That's you're goal. If you have less common problems, put them into the design techniques which are already there and you can do research using part of these expert systems.

Now there are some limitations that I thought of. Our knowledge is limited because what we have is limited. Most important, I don't think there is an expert system to learn from itself. But what will happen to all the gurus, like we are? So this is what happens to all the gurus, like we are. We go to workshops. [laughter]

SHANMUGAN: Thank you for being on time Phil. Any questions for Phil? This is really a list of wishes that one would like to see in an intelligent CAD tool whether it is at transmission systems level, or at network level. As Phil pointed out, we are not there yet and hopefully in the next 3 to 5 years we will get there yet. Frank do you have a question?

CHETHIK: Well as an observation, wouldn't all of the novice engineers that use these expert systems stay novices? Would they not gain any insight?

BALABAN: No, I don't think so, because they have the diagnostics, as to how the decisions made was available to them. And my advice to you would be, why don't you go and read Shanmugan's book or something? That could be one of the things. But as to the wish list, these are only for routine problems, I mean about 80%. Also, some of the problems like Mike had, where you have one system which has been designed for 5-6 years, is different. You have one simulation system which you use over and over again. But if you have a smaller system, with fast prototyping, this would be ideal for that.

AMOROSO: I have a question, and I also have possession of the microphone, I'm over here. I'd like to know if anyone has ever published a perfectly general definition of an expert system?

BALABAN: I don't know.

MOUFTAH: There are several people who have defined an expert system.

BALABAN: If you can, for any problem create a tree which has an answer without searching and asking, that is a program problem, that's not an expert system. An expert system which is more ambiguous since it has a lot of rules of thumb and the system is going through these rules to make a decision, sometimes it is also deterministic. If for example you have weather forecast, it is a very complex system. And in order to forecast the weather for tomorrow, it takes you five days, and so it is not very good. But there are rule of thumbs to have more or less. That's an expert system too, even for deterministic problems.

SHANMUGAN: Thank you, Phil. As I mentioned earlier, because of a broken cable, we will not have a demo of BOSS. We are substituting a 10-minute presenta-

tion on the Pink-Jeep tour in place of the BOSS demo. [laughter] BOSS stands for Block-Oriented Systems Simulator and my presentation is not aimed at a sales pitch for BOSS or anything like that. But I wanted you to have a good feel for where we are by way of current state-of-the art, in terms of CAD tools for transmission systems engineering on communication link design.

This started as a giant effort between the University of Kansas and TRW. Besides myself, the people who were involved in this project were Ed Kahn and Gary Minden. At TRW the prime movers at the technical level were Tom Manning and Greg Wiswall. We also had the blessing of the management in the form of Raul, Dick Booton and everybody else at TRW on this project. There are a number of CAD tools that are currently available for link level analysis and design. Many of the earlier tools were strictly formula-based. These were programmed and written to do things like link budget calculations. They really did not do detailed analysis of the functional block in the communication system. Later on starting with SYSTID in the late sixties, early seventies, a number of CAD tools based on simulations, detailed waveform level simulation work developed, and then we got into hybrid approaches that combine some formula-based analysis with simulation-based analysis. Good examples of hybrid tools, BOSS is one, COMSIM and SYSTID to some extent are hybrid CAD tools also.

BOSS, as I mentioned stands for block-oriented system simulator. In very simple terms it can be viewed as an operating system on an environment for doing simulation-based modeling and analysis of a communication system. So in some ways it is like DOS (disk-operating-system) on a Fortran compiler. It permits the communication systems engineer to represent the system, whether it is a unit, a subsystem, or a top-level view of a complicated system, in a hierarchical block-diagram form. So right there BOSS differs from other systems in that the engineer, after he is describing a system for simulation-analysis purposes, uses the language that is natural to him, namely, block diagram representation. BOSS takes over from there and BOSS can be used to configure systems in any arbitrary topology and any arbitrary block-level block diagram type of representation. BOSS takes over from there and does a waveform level simulation of the system, and presents the simulation results, and permits the engineer to perform design iterations. There is a clear separation of responsibilities in BOSS. We assume that models in subsystems can be represented in an hierarchical block-diagram form. The communication engineer is responsible for the correctness of the block-diagram representation and for the the correctness of the parameter values. The software responsibilities are assumed by the software package itself. A lot of times the comm. engineer who is usually a very thorough programmer has to assume responsiblities for programming too. And that is where most of the errors come in most of the time. In BOSS, there are separate responsiblities. The engineer is responsible for describing the topologies of the system and the parameters correctly. The software package itself is responsible for all other functions.

Let's say someone is interested in analyzing a system like the one I've shown on the top portion of this viewgraph. Suppose you are interested in looking at what happens if you put a higher-order signaling format like 16-qm through a non-linear satellite channel. If you are interested in

evaluating the performance of a system like this, and if you're trying to optimize certain bandwidths, cut offs, cost of the nonlinear amplifier and so on, you will use BOSS in the following fashion to address this problem.

First of all, BOSS is a manual driven system, and to construct a system model, you'll use BOSS and the BOSS menus to actually draw a block diagram on the screen. You draw a block diagram representation of the system on the screen by simply selecting functional blocks that already exist in the BOSS database. For example, I have a random data source that already exists in the BOSS database, so I call it by simply going to the menu of digital sources. From the menu of digital sources I select a random data source, and the block diagram representation appears on the screen. The block diagram representation has the name of the block. It shows signal ports, output signal ports, and if you have input signal ports, they will be shown explicitly. It also shows a little rectangle here indicating that there are certain parameters that the user can set, or there are certain parameters that you can iterate on, or there are certain parameters that you can actually hook up from machine data.

So the first thing you do in using BOSS to construct a unit level model or a subsystem model, or a system model, is to bring in the functional blocks that you need to represent the system on the screen. Now, if a functional block does not exist in the model library -- for example, if you are looking for a particular type of nonlinearity which you want to put in the system somewhere, and a model does not exist -- you can construct that model using what are called BOSS primitive building blocks. Primitive building blocks are simple things like adders, multipliers, limiters, and so on. By putting these together you can construct a model of an arbitrary nonlinearity that you might want to put into the system. Or if that's not adequate, BOSS gives you a provision to actually write your model in FORTRAN and plug it in the block diagram representation, into the BOSS environment.

Once you have all the functional blocks, the next logical thing to do in BOSS is to make the connections. Connections in a block diagram representation in BOSS are made simply by pointing to the input port and next pointing to where it needs to be connected. BOSS automatically draws the connection for you, and will keep track of the signals.

The signal types in BOSS are quite numerous. You can go from logical signal types to real valued, integer, complex and even vector-type signal values will be handled by BOSS. And as I mentioned, all of the signal type setting and so on are automatically done by BOSS or the user. If you have a complex-valued signal, and you connect it to a real-valued input port, BOSS will immediately stop you and say that the signal types are not consistent.

Also BOSS provides on-line help all the time. Anywhere, in any part of the process, if you're lost, you can push the middle button on the mouse and help information will appear on the screen automatically. Appropriate help information will appear depending on where you are in the block diagram, like if I ask for help by pointing here, I'll get the help on this entire module. If I ask for help by pointing to the signal, I'll get help information on what type of signal it is, what its ranges are, and so forth. If I point to here, and ask for help, I'll get a description of all the parameters of this block, how to set parameter values

and so on. So there is a tremendous amount of consistency checking and intelligence in the form of on-line help and documentation built into the BOSS system. BOSS will also ask for and will do on-line documentation.

Once a block diagram is completed, or even if the block-diagram is not completed, you can go and set the parameter values. BOSS offers the maximum flexibility when you are getting ready to set parameter values for an analysis or simulation. Numerical values can be entered for any parameter. In this example I'm setting the filter order for this particular filter to be second order. So in all the simulations that will be run, the filter order will be set at 2.

Now I could also go to, for example, a datafile in the BOSS database and specify an arbitrary frequency response for this filter. In choosing this to be a Butterworth filter, I can go and choose the frequency response contained in such and such datafile. That frequency response could have come from a unit that's already built and you are using in the simulation. Also, any parameter that appears on the list can be exported -- exported means you float it to the top level -- then you can specify iteration values for the particular parameter. For example, if you want to look at the effect of the receive-filter bandwidth or the transmit-filter bandwidth on a system performance, you can iterate through several values automatically. And at the end of the simulation BOSS will plot, for example, error rate as a function of transmit-filter bandwidth, receive-filter bandwidth, etc. So you can do all the design iterations automatically rather than making multiple runs and plotting it by hand, or going to a separate plot package.

If you do everything right, when you are done with the block diagram representation, you go into the simulation phase. What this viewgraph shows is the hierarchical representation that is used in BOSS. At the top level I have a 16-level qm source. Even though it is shown as a single functional block, it has a lot of details inside it. So if you are the engineer who actually built the 16-qm modulator, you'll be dealing with one or two levels below in the hierarchy, and you can have any level of detail that you want. For example, in the 16-qm modulator, all I'm showing at this level is that the source is made up of actually four binary sources going into the 16-qm modulator. This modulator itself may be made up of two QPSK modulators with appropriate gains in phase shifts. The QPSK modulators themselves may have been made up of two binary PSK modulators with appropriate gains, phase shifts and so on. So we allow the user to construct the model with any level of detail that he wants and also view and simulate the model at the level that is appropriate for the particular engineer. So if you're a unit engineer you will actually simulate this particular block in all of its detail and glory. Whereas if you are a systems engineer, this is just one functional block of the system.

These little pointers here are the analogous of alligator clips that you attach to test points to view waveforms. BOSS uses these pointers to collect waveforms that you can analyze later on. And once the block diagram is complete, then you set the simulation parameters and go into simulation and then look at the simulation results in a variety of formats.

This is one screen from the BOSS workstation. The block diagram is shown in the middle and you can look at the simulation results by simply pointing to on the block diagram and asking for a signal on any of these points. Then BOSS will

CAAD TOOLS FOR ADVANCED COMMUNICATION SYSTEMS ENGINEERING

- Good systems engineering CAAD tools are needed to: create new, more reliable, high quality products, quickly and less expensively in small lot sizes

- Reduce time required to do systems engineering of complex products (rapid prototyping)

- Reduce time required to insert new technologies into products

- Reduce the cost of systems engineering

- Improve the accuracy of design

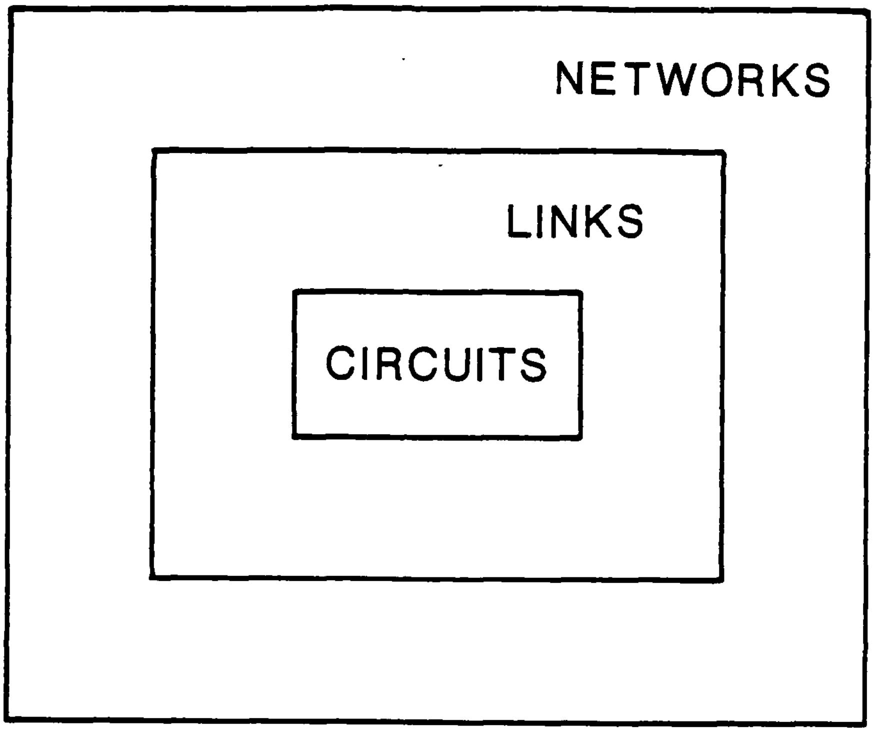
SLIDE 1

COMPUTER-AIDED ENGINEERING TOOLS

RESEARCH	Mathematical Modeling and Analysis
	Rapid-Prototyping
DESIGN AND DEVELOPMENT	Computer-Aided Analysis
	Simulation
	Computer-Aided Design
⋮	⋮
MANUFACTURING	CAM Tools

SLIDE 2

HIERACHICAL VIEW OF CAAD TOOLS FOR COMMUNICATION SYSTEMS



- A large number of CAAD/CAM tools are available at the circuit/chip level
- Not many standard tools are available for systems level (link and network) analysis and design
- Lack of integration between CAAD tools at various levels

SLIDE 3

CAAD TOOLS FOR TRANSMISSION SYSTEMS

- APPROACHES
  - Formula-Based
  - Simulation-Based

- SIMULATION-BASED CAAD PACKAGES

  SYSTID
  COMSIM
  BLOSIM
  ICS
  ⋮
  BOSS

- CURRENT TRENDS/FUTURE DIRECTIONS
  - Improved modeling and simulation techniques
  - Modular software packages with graphical representations of building blocks/postprocessor
  - Expert systems framework

SLIDE 4

FUTURE DIRECTIONS

- Developing computationally efficient simulation models
- Improving the computational efficiency of Monte-Carlo simulations
- Developing an expert systems framework to guide the analysis and design of complex communications systems
- Integrating network simulations, link simulations, and circuit simulations
- Top down design features

SLIDE 5

present the simulation result in a time-domain, frequency-domain, or any other kind of statistical display like histograms, correlations and so forth. All of it is done without ever writing a single line of code if all the modules you need exist in database.

You're simply selecting items in the menu, drawing a block diagram literally on the screen, having the system take care of all the software functions, doing the simulations, looking at the simulations of the software. You don't even have to remember the name of any of the signals because you can bring the block diagram up and ask for a signal plot by pointing to it. It is hierarchical, so the levels of details can be hidden from the user whenever it's appropriate; on the other hand if you do want to look at the levels of details, they are available to you. It is self-documenting to a large extent in that any time you build a module, you have to document the module as you're building it. That information goes into a documentation database, it also goes into an on-line help database, too.

At the end of a design, you can ask for complete documentation, and BOSS will provide you block diagrams of each functional block, all the way down to the lowest level, and all the documentation, too. So it is a complete analysis and design tool and it is kind of representative of where we are in CAD tools for transmission systems engineering.

This is about a one-hour talk which I am trying to condense into about 10 minutes so I am skipping a lot of details. A good reference for BOSS is the paper that I gave at MILCOM'86. If you look at the *Proceedings of MILCOM'86*, there will be a detailed paper on BOSS and if you need any additional information, I'll be glad to send it to you and if you have any questions, I will be glad to answer them for you right now.

REIFFEN: What is the computer system that it runs on?

SHANMUGAN: Right now it runs on a DEC VAX station. We are in the process of converting it to run on SUN and APPOLLO workstations, too. At the speed of this workstation, if you are simulating a system like this, in real time you can simulate something like 10 symbols per sec. That's the speed. The processor speed itself is 1 MIP, but the simulation speed is like 10 symbols per second of real time.

REIFFEN: Could you give us some sense of how rich the various functions are that have been developed? Could you give us a snapshot of where it is and what additional work needs to be done in order to make it a general purpose, as you would define it?

SHANMUGAN: It is a very general purpose framework in that you can use BOSS to do control systems simulations now. Anything that you can represent by way of modules .....

REIFFEN: No, I am not asking about the concept, I am asking about the menu of preprogrammed functions.

SHANMUGAN: I'll show to you, for example, a snapshot of the groups that we have in the model library, there are preprogrammed modulators, analog sources, channel models, demodulators, and so on. Now BOSS will not have all the models that you need for all applications. To some extent that expert is precise in the various organizations that are using BOSS. What we are providing is a very easy to use framework for you to construct your own model and put it in the model library, and use it immediately. We don't make any distinction between BOSS-supplied models

and user-created models.

WELCH: In other words if somebody has built the directional demodulator, will that appear in your menu?

SHANMUGAN: That will automatically appear. As soon as you put it in the library, it will automatically be added to all the menus at the right levels. So you really ought to look at it as a tool for building your own model library. You don't have to come to me or any other guru to build BOSS models. You can build it with what we provide to you, or you can build it in the form of your own FORTRAN primitives, add them to the library and go on from there.

REITMEYER: Do you have the right functional models in the language?

SHANMUGAN: At the lowest level the models are written in FORTRAN. Once we have a low level FORTRAN model, from there on it can be used in a block diagram representation at the higher levels in the hierarchy. It is only at the lowest levels that you write FORTRAN.

SPILKER: An obvious question is what's the availability of BOSS to industry and to other universities?

SHANMUGAN: It is available to the public domain. There are 15 installations of BOSS already, and there are no restrictions on the distribution of it. There is a price that goes with it.

REIFFEN: What is the price?

SHANMUGAN: You can talk to me about it or you can talk to TRW. The price is less than $10,000 for industry, $2,000 for the universities. That's one of the advantages of developing a tool like this in a university environment where there are not too many restrictions on proprietor rights and so on. TRW was very gracious in terms of giving us permission to release it in the public domain.

REIFFEN: Could you give us some sense as to how long it took to develop it to its present status and effort?

SHANMUGAN: It took about 2 years from the date we thought of developing a block diagram oriented simulator. It took us about 6 months of design before we wrote the first line of code. There are about 4 people working on the design, so about two man years of design went into it. The entire package except for the models is written in LISP. That language offered us a lot of facilities that are not available in other languages. So after six calendar months and about two man years of design, it took us another six months and about an additional two man years to develop the first working version of BOSS. Then it took us another calendar, and about three man years to develop the production version of BOSS. So it took two calendar years and and about six to seven man years to put it out on the market. Any other questions?

CHETHIK: When will it be ready to run on a SUN workstation?

SHANMUGAN: In about approximately three to six months. The SUN version will be out in three months, the APPOLLO version in six months.

WELCH: Suppose that I had a microvax and an array processor, is there any way to pipe some of the stuff over to the array processor, and then send it back?

SHANMUGAN: Certainly. There are operations, like FFT filtering operations, that you could conveniently ship to the array processor. For other operations, since this is a time-step oriented simulator, you will not save anything by going over to the array processor. But when you do block processing like FFTs, certainly. Any other

questions or comments?

I think it is time to take a short coffee break. The next part of our session will be devoted to CAD tools for networks. There are three speakers and it will take about an hour to go through those presentations. So let's take about a 5-10 minute break and start again.

BREAK

SASTRY: My presentation is based on work that we did in the past five years (SLIDE 1), building various network models, each of which is approximately a man year type of effort. In building these tools, we have been using SIMSCRIPT which is a discrete event simulation language. There are many other building tools like SIMULA, GPSS, things like that, and also these days some icon-based graphical tools are also coming out like SIMKIT, which is a LISP-based one; Kurose is going to say some more things about those.

SLIDE 2 gives a list of the models that we built. Don't be afraid, I'm not intending to go through each one of them, there's no time. The purpose of this talk is, based on that experience, to present the sort of problems we were faced with. Specifically, the distributed processing in networks, which we are currently very much concentrating on, and a few topics like how to reduce simulation run time and things like that.

When it comes to distributed networks the problems we find are shown in SLIDE 3. A realistic workload characterization in a network is extremely important, because you may have a very good model of the network, but the results that you get are only as good as the traffic model that you put in. Using a Poisson model for the traffic is fine, but in a real world simulation, your simulator has to give you answers for a real world network which you are planning on, and then there would be considerable departure from what you find and what you expect. And since we are now talking about simulation, which is supposed to be giving you additional results other than what you can get through queueing theory, you would naturally like to put in some effort on defining more clearly, what the realistic workload characterization is. And one of the tools that we should really have is how to define what's going on. Supposing you look at, say, the OSI (open system interconnection) layers. It is a seven layer model for the network, we are talking about the top three layers. They are essentially application oriented type of environments from which you should get the workload characterization. We have developed two procedures for doing that. I'll just briefly describe to you one of them.

Other important issues related to the simulation of networks are the interlayering interactions among these seven layers and the transient bottlenecks, as we have heard this morning from Professor Turin. I was a little worried to mention these things earlier, but now I feel more confident since he mentioned that! We are also interested in fault management. We have been doing some work on that and in the process we found the most difficult part there is the simulation of faults. We have developed some message-based fault-isolation algorithms, to try to track down errors in the network. But, what sort of errors do you expect in the network? At the hardware level, the software level, or the protocol level? Simulation of those faults is very important. It is difficult to do, and this is, in fact, the most difficult problem of all.

For example, if you are looking at a real time simulation run, suppose you have a distributed computing system, and you

want to simulate a half an hour of real-time usage of that computing system. A half an hour of that simulation at the message level, or at the packet level, involves several millions of packets. And if you're having a discrete-event simulation, the time scales are so different than at the application processes level and the packet level, you'll be creating so many events, that half an hour of real-time might end up as, actually, a simulation run time of say 24 hours or even 2 days. And now, imagine the complexity, if you want to take it to the bit-level detail, as Sam and some others are doing. Obviously, you cannot start from the bit level and the packet level, and continue all the way up to the application process level having a single overall simulation model, without filtering out some of the details to reduce the complexity of the model. If you are satisfied with the average results at the link level, then you have to take the average parameters, but if you're not satisfied and you need second-order statistics, you have to collect them and then plug them into the models. So the models have to be built in a layer based structure, if you want to keep the run times within reasonable level. Now I'll show to you something about the results.

Here is one approach to workload characterization (SLIDE 4). What we did was, for the top three layers, define various tasks that are involved in a distributed processing system. And if you know those tasks and you start this task, for example, message 1 will start task 1, which will create some additional messages, and that will go to task 2, maybe then, one of the messages from task 2 may open up task 3, and then 4, but they need not be in a sequence. For example, 4, 5, and 6 may immediately go to task 1 again. So these tasks are interlinked by corresponding messages. Hence if you can identify them in a distributed system (here again, this is an example), and if you can link up these tasks with a proper message model, then you don't have to describe the individual messages the tasks will create. The creation mechanism has to be incorporated in the task. These tasks could be totally asynchronous, independent tasks, or they could be subtasks of a larger job. So we have a very flexible model, which took us about a year to develop.

This is written in SIMSCRIPT, and there are lots of user-selectable features. This is one particular model where you have tasks and messages. We have another model which you can use, in case you know the entire sequence of messages that take place in a system. We have simulated a newspaper production system, for example, using local area networks, and the entire control message sequences that take place in a newspaper production system. Since we know the messages and the timings that these messages were created at, for example, all these things can be put in a proper structure. In fact I don't have time to go into that. But the structure again, is not rigid, i.e., the number of combinations, such as how many printing presses you want, how many types of jobs run, which might also be overlapping, possible phases in the jobs, all these details can be put in. So with some input parameters, you would be creating real jobs, really running, and they would create all the messages. These are the two types of models that we have but I don't claim that this is the only way to accomplish that, there are many other ways. I thought of mentioning just this one, so that many others here could perhaps think of creating a realistic traffic model.

This is especially important for networks because, looking at the OSI model,

from each layer to layer down, when you have these peer entities talking to lower entities, there is what is called mixing taking place, i.e., several entities at the top layer might talk to a single entity in the lower layer, in which case it is multiplexing, or, one entity in the top layer might talk to several entities in the lower layer, in which case it is called splitting. So you have this splitting and mixing taking place, and by the time you come to the data link layer, which is of interest for example to the earlier presentations, you lose all correlation between the source and its corresponding message, so that you finally have a bunch of independent messages. Thus at the data link level, usually it is appropriate to use a Poisson model, but if you go to a higher layer, for example if you want to do a transport layer performance, you should know more precisely what is the type of traffic that you are really creating.

Here we have a system called simulation analysis of distributed systems. This is the program which takes in those traffic models, as well as at each other layer whatever type of protocol that you have, as the bus-based protocol used in CSMA CD or token-passing bus, or whatever. This is the total system and this is how the outputs look (SLIDE 5). You can have plots for example, for the resource utilization, link utilizations, mean-message queues, and things like that.

In addition to that, to get an idea of how things go on with transient features, I have one slide here for you (SLIDE 6). It is an off-line system, and as the simulation keeps running, we take outputs at selected intervals. In this case, a 20 sec simulation was taken and then we got outputs at every second. So you can see what the output looks like at every second. What is shown on this slide is the state of the system, for example the CPU utilization. The current one is the upper block, and the previous one is the lower block. The lower block is green and the upper block is white. Green means 20-40 utilization, white is 0-20. So as the simulation keeps running, though what we are seeing is not in real time but off-line, you will see the colors jumping between here and there. You know the previous state and the current state. So you get a picture of what is really happening during the transient situation. To give you an example, let me choose (results for node) 6 here. The 6 will show you, the lower one is green, i.e., 20-40, and finally at the end of the 20 seconds, it shifted to white. That's where it ended, at the lowest utilization. A corresponding black and white version that I have is for 3 seconds instead of 20 seconds. You see (for I/O processor) at 6 both of them were heavily utilized. They are actually at 80-100 level. At some point they are 3rd level. So if you go right away to the end of the simulation, you will not see what is really happening during the simulation time. So there are situations where you can pick some of the transient, bottleneck problems. But how do we interpret what those things really mean, that's a completely different question.

We have been looking at some of these issues, both transient analysis and what kind of statistics we should get, how to handle priorities of messages and things like that with particular reference to token-passing bus. Most of you may be aware that token-passing bus is a local area network scheme which is part of the manufacturing automation protocol (MAP) that many people now think is part of an ipso facto standard for factory automation (SLIDE 7). General Motors initiated that and many others have accepted it now.

In the seven layers there are various other protocols. Previously it used to be in the second level, but now the token passing bus is coming to level 1. And we started working on token-passing bus before the MAP came in, about 3-4 years ago. The token-passing bus has several features in it (SLIDE 8), I don't want to go into all of that, except pointing out that there are 4 level priorities which is a facility, but it also makes things very difficult to understand the system.

I just wanted to show you this interesting feature. You don't have to look into all the curves (SLIDE 9) except one, the one with the average delay per frame. Here the mean-interarrival time is actually increasing, as mean traffic level is decreasing. Initially the traffic is very high (low mean inter-arrival time), and there is a particular delay. As traffic is decreasing you would expect the delay to improve. It actually improves, but it starts rising at this point and then falls back again. The reason why this happens is that at this point, only high priority messages are being transmitted; lower priority messages are not getting a chance to be transmitted. When they start getting a chance at this point, each lower priority packet that was transmitted carries with it a large queueing delay, so it adds to the average. But eventually it comes down. So if you just look at this curve, it is very difficult to interpret what is happening to the overall network. Obviously you should have delays measured for each priority level. So basically what I wanted to say is that if you have priorities for the messages, it can muddle up the interpretation of the results very much.

The run-time reduction is another problem. I just mentioned earlier the time factors (SLIDE 10). They are different between the application processes and the lower level communication processes, and they can blow up the run times. A couple of solutions for that could be that wherever possible, compute the impact of events rather than simulating every event. For example, token-passing bus has several stations connected to a bus and a token goes around. If some stations don't have messages to transmit, they don't use the token, they simply get the token and pass it on. But if we know how many stations have queues without any messages in them, or rather, have buffers without any queue, then you don't have to really simulate those events, i.e., the token being received and sent again, and things like that. Instead of that you can compute the time it takes for these events to take place, and then update the simulation clock. By doing that you will save quite a bit of time, and we have done that for the token-passing bus. But it is a very tricky process because when you update the simulation clock, you must make sure that no other events are lost, like a new packet created, or a packet coming in from somewhere else, etc. So you have to scan for the next scheduled event. It is a fairly involved process.

The next issue is related to distributed simulations. This is the case when you are allowed some sort of parallelism in your simulation, then you can perhaps go on doing that in several machines at the same time, or several processes at the same time, and interconnect them through messages, which is difficult because you have got to keep synchronization among these processes. This is quite involved, but still it promises to be a solution. We are working on it, and it is catching up very much. Quite a few others are also interested in that.

Now the next thing is the validation of simulation results. This is a real problem.

When steady state solutions are required, perhaps people can use confidence intervals (SLIDE 11). But most of the network solutions that we are looking at are really very dynamic in terms of traffic changes. The question is can you really get steady state solutions or not? That is difficult to answer. But we are also definitely interested in some transient situations. How do you validate transient results? How do you know that you are getting an odd and fictitious value or something that is not dependable? You want transient results, but at the same time, you really don't know how to validate them. I'm not talking about validating the model, I'm talking about validating the results.

There are some other problems that you might have, as for example you want a distribution of the delay, instead of the average delay. You can evaluate the average delay analytically. But since you are going through the trouble of getting a simulator, you should get a distribution of the delay. But in order to get a distribution, what happens is that many samples fall away from the average if your standard deviation is very large, and then you cannot really get accurate values for the samples. By using some extreme value theorems or something like that, people may be able to find some tools in trying to approximate the distribution.

Then what will happen to the queue sizes, suppose you want realistic queue sizes, how long do you want to run the simulations? We have done some experiments. For example, here, as a function of simulation run-time, we have tried 3 different initial queue sizes, and tried to run them (SLIDE 12). And as you can see, one of these values gives you sort of a periodic thing. And the other one is trying to settle down. This one takes a lot more time, higher peak and lower peak, so you may have to wait for a longer time. Sometimes it is worthwhile to choose an efficient initial value rather than starting from the zero, some extremental value. In fact you can see another curve where we have started at the same value (SLIDE 13) but this time we plotted the same initial queue size but for different traffic levels. And you can see here it is getting to be unstable, the queue sizes are building up. The traffic is actually low, while here it is almost periodic. So there are some situations where you can see that you are really saturating the network. Some such conclusions can be drawn.

Let me now introduce another major problem that we have been facing (SLIDE 14). Suppose we take a multi-hop packet radio network, i.e., there are several mobile packet radios (in fact, we also use the same thing for multiple satellites networks), what is the throughput in a network? Throughput in a single-hop network or a common bus is well understood as the useful messages that have been delivered to the receiver. Now here, how do you evaluate that, and what is throughput? It is very difficult to understand. But in a simulation at least you can use some sort of a yardstick like, call 'traffic' as all transmitted packets, which you can count in a simulation and call as 'throughput' all successfully received packets, and then take the ratio and call it 'utilization'. But still it is defined on overall network level. If you have a single transmitter and many receivers (broadcasting), what is the throughput in this case? That is a very critical problem. I think we still have to get some very good parameters to describe this sort of system.

Here are some results, familiar results, i.e., how the utilization varies with or without retransmission, the mean-interarrival time delay, these are all average

statistics which are reasonably understood. We can even get these results perhaps from queueing models. Hence the selection of the right parameters will facilitate an effective use of simulation models. It is not enough just to build the models, a very important issue is how to use them. That is what I've tried to summarize. If I had more time, I could have described this further.

Basically, to recapitulate some of these things (SLIDE 15), the network simulations can provide additional insights, especially in the case of handling priorities, routing, and getting delay distributions instead of average delay. That is why network simulations are particularly helpful. And then more realistic traffic models can be constructed and used rather than Poisson models, as well as the transient behavior that I've already discussed, things like bottlenecks, can be better understood. With just a single bottleneck, suppose some buffer goes out, and because some protocol fades, the whole thing falls back, which will not be shown in an analysis of average results. So you really need to look at these bottlenecks.

Time-dependent traffic is a very important parameter that can be observed. Interpretation and generalization of results have pitfalls. That's where the need for validation of simulation results and other things discussed earlier become important. There are certain analytical tools that can aid simulations. I am not talking about parallel analytical tools in which you can say, well, our simulation confirms the analytical results, that is not what I'm talking about. To get a good successful simulation model, and to be able to use this you need some analytical tools to represent some processes and to evaluate results.

That's why it is some sort of a hybrid thing. As I was mentioning earlier, priority-handling algorithms have to be developed. For example, in the token-passing bus, there are so many stations, each station has what is called a target token rotation time, and a change in any value will affect everybody else's transmissions. Some people tried to represent it as a sort of linear programming problem, there can be many other approaches, but that is a very important problem. It has not yet been solved. People talk about factory automation, but the first thing they find later is how you fix and initialize these values, and how to dynamically alter them as the traffic keeps changing. You do need quite a bit of analytical work there, otherwise you may do 500 or 1000 simulations, like wild-goose chasing, you'll never get a right solution. So you will have to narrow down the problem by using some analytical algorithms first, and then fine tuning using the simulations. Then estimation of distributions based on small sample sizes can also use some analytical work. Then of course distributed simulation may be useful, and we are trying to work on it. Basically the problem is how to keep synchronization among various parallel operations. Sometimes the simulation itself is not amenable to parallel operations, then of course you cannot conduct a distributed simulation.

I don't know if I'm permitted any questions?

SHANMUGAN: That's the whole purpose of the workshop, is to have questions and discussions.

PICKHOLTZ: You said you use SIMSCRIPT. The same people who developed SIMSCRIPT, CACI, now also have a specifically designed network simulation tool called network 2.5 which in fact

NETWORK SIMULATIONS: (*Lack of !*) TOOLS FOR EVALUATIONS

A.R.K. Sastry
Rockwell International Science Center
1049 Camino Dos Rios
Thousand Oaks, CA 91360

- INTERACTION OF PROCESSING AND COMMUNICATIONS IN DISTRIBUTED NETWORKS
- REDUCTION OF RUN TIMES
- INFLUENCE OF MESSAGE PRIORITIES
- VALIDATION OF SIMULATION RESULTS
- PERFORMANCE CHARACTERIZATION OF MULTI-HOP NETWORKS

SLIDE 1

Rockwell International
Science Center

SIMSCRIPT–BASED SIMULATION MODELS DEVELOPED DURING 1982–87

(Rockwell International Science Center)

- MESSAGE-SWITCHED STORE-AND-FORWARD NETWORK
- CIRCUIT-SWITCHED NETWORK
- AUTOMATIC CALL DISTRIBUTORS
- TOKEN–PASSING BUS LOCAL AREA NETWORK (IEEE STANDARD 802.4)
- FDDI TOKEN–PASSING RING LOCAL AREA NETWORK (ANSI X3T9.5)
- TRANSPORT PROTOCOL (TP–4, ISO STANDARD)
- SIMULATION ANALYSIS OF DISTRIBUTED SYSTEMS (SADS)
- MULTI–HOP PACKET RADIO NETWORK
- MULTIPLE SATELLITE COMMUNICATION NETWORK

SLIDE 2

Rockwell International
Science Center

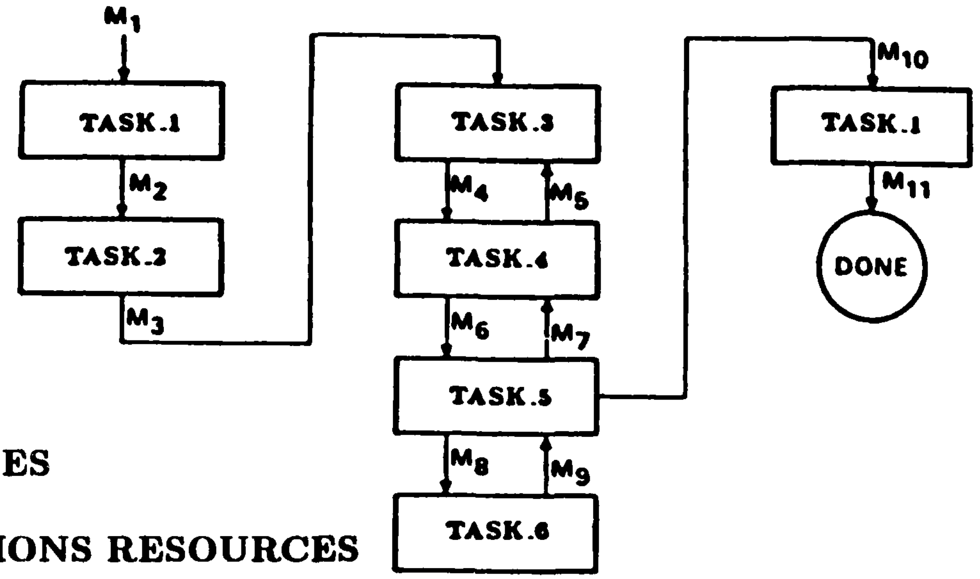
PROCESSING AND COMMUNICATIONS IN DISTRIBUTED NETWORKS

- REALISTIC WORKLOAD CHARACTERIZATION
- REPRESENTATION OF INTERLAYER INTERACTIONS
- TRANSIENT BOTTLENECKS CAN BE CRUCIAL
- FAULT MANAGEMENT EVALUATION: SIMULATION OF FAULTS
- DIFFERENT EVENT-TIME SCALES AT THE PROCESSING AND COMMUNICATION LEVELS CAUSE SIMULATION RUN TIMES TO BE VERY LARGE
- PRESENTATION AND INTERPRETATION OF RESULTS

SLIDE 3

Rockwell International
Science Center

WORKLOAD CHARACTERIZATION IN DISTRIBUTED PROCESSING NETWORKS

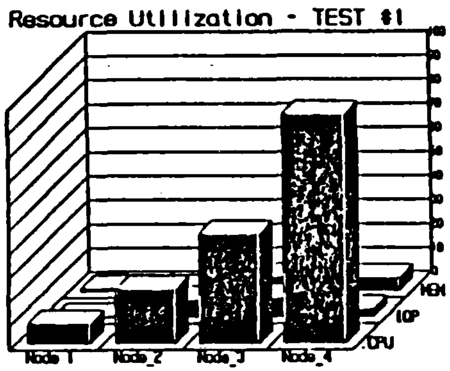


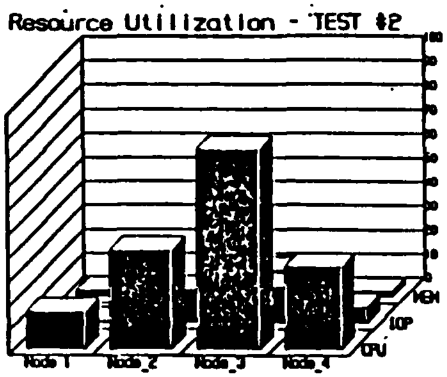
- MESSAGES INITIATE TASKS
- TASKS CREATE MORE MESSAGES
- MESSAGES USE COMMUNICATIONS RESOURCES
- SIMULATION ENDS AFTER THE FINAL MESSAGE
- EVALUATION IS BASED ON UTILIZATION OF RESOURCES AND RESPONSE TIMES

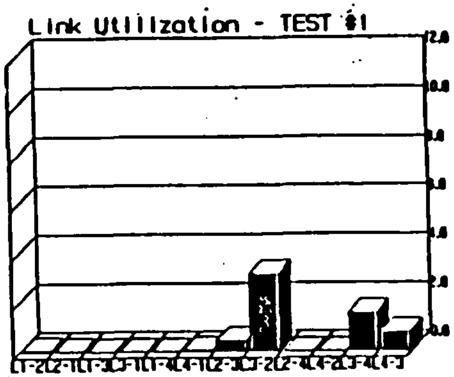
SLIDE 4

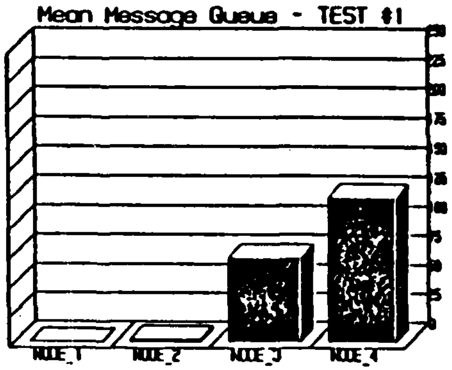
Rockwell International
Science Center

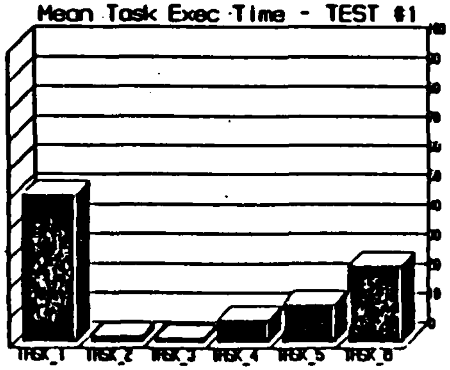
SIMULATION ANALYSIS OF DISTRIBUTED SYSTEMS (SADS)



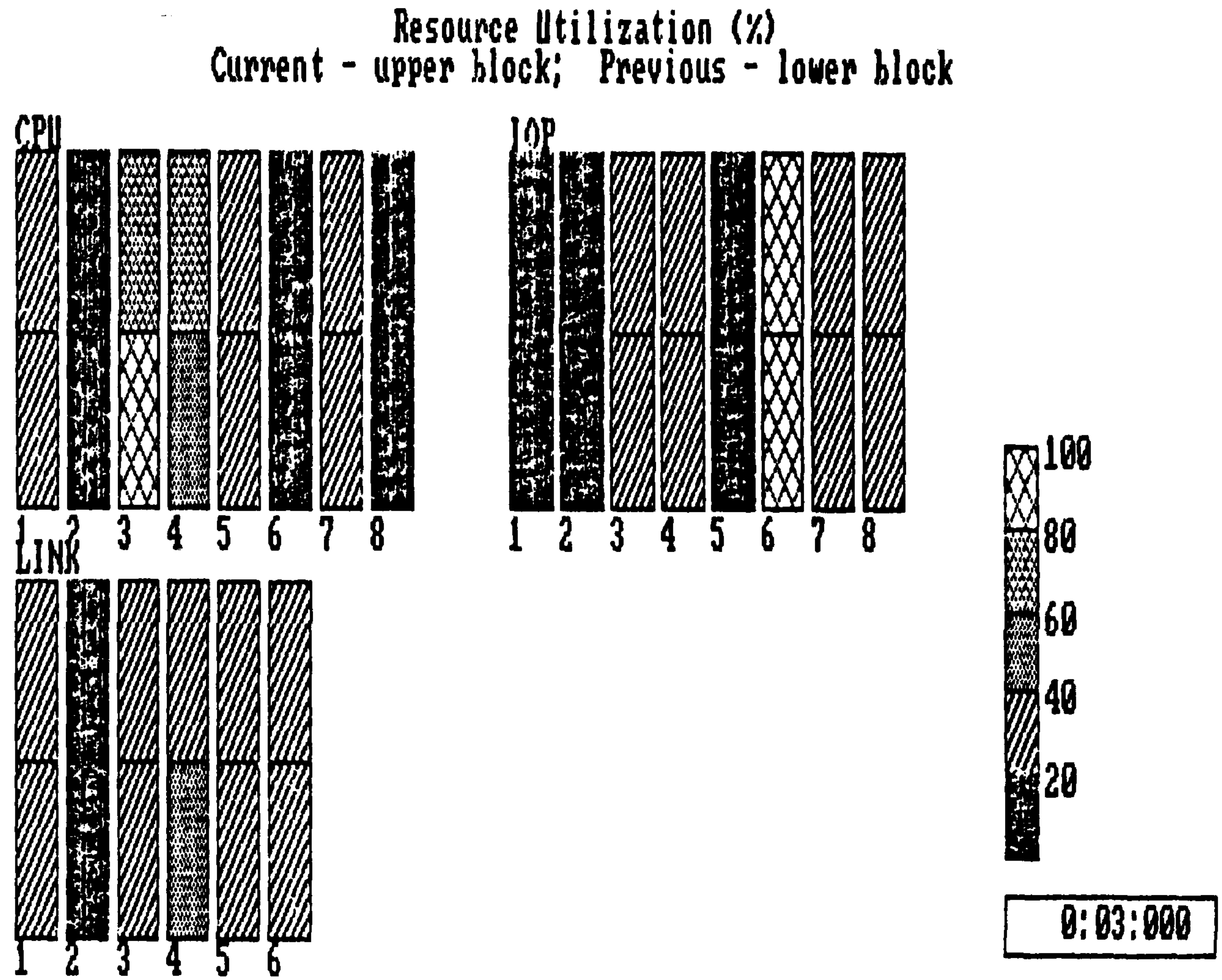








SLIDE 5

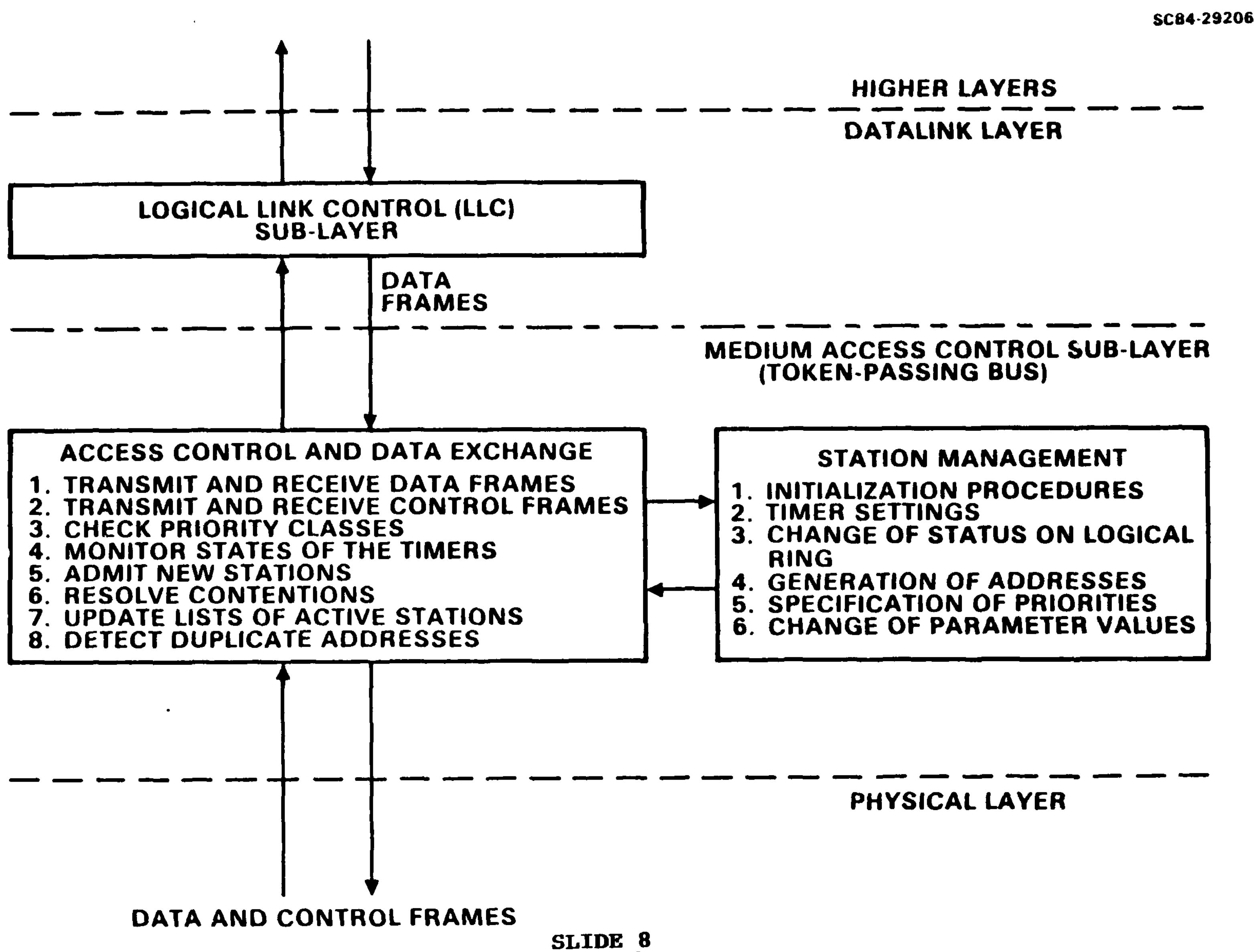

Resource Utilization (%)
Current - upper block; Previous - lower block
CPU
1 2 3 4 5 6 7 8
IOP
1 2 3 4 5 6 7 8
LINK
1 2 3 4 5 6
100
80
60
40
20
0:03:000

SLIDE 6

MAP 1985

LAYER	MAP VERSION 2.1 PROTOCOLS
7	ISO FTAM (DP) 8571 MANUFACTURING MESSAGING FORMAT STANDARD (MMFS) COMMON APPLICATION SERVICE ELEMENTS (CASE)
6	NULL
5	ISO SESSION (IS) 8327
4	ISO TRANSPORT (IS) 8073, CLASS 4
3	ISO INTERNET (DIS) 8473
2	ISO LOGICAL LINK CONTROL (DIS) 8802/3 (IEEE 802.2) TYPE 1, CLASS 1
1	ISO TOKEN PASSING BUS (DIS) 8802/4 (IEEE 802.4) (ISO IS CONSIDERING MOVING IT TO LAYER 1 FROM LAYER 2)

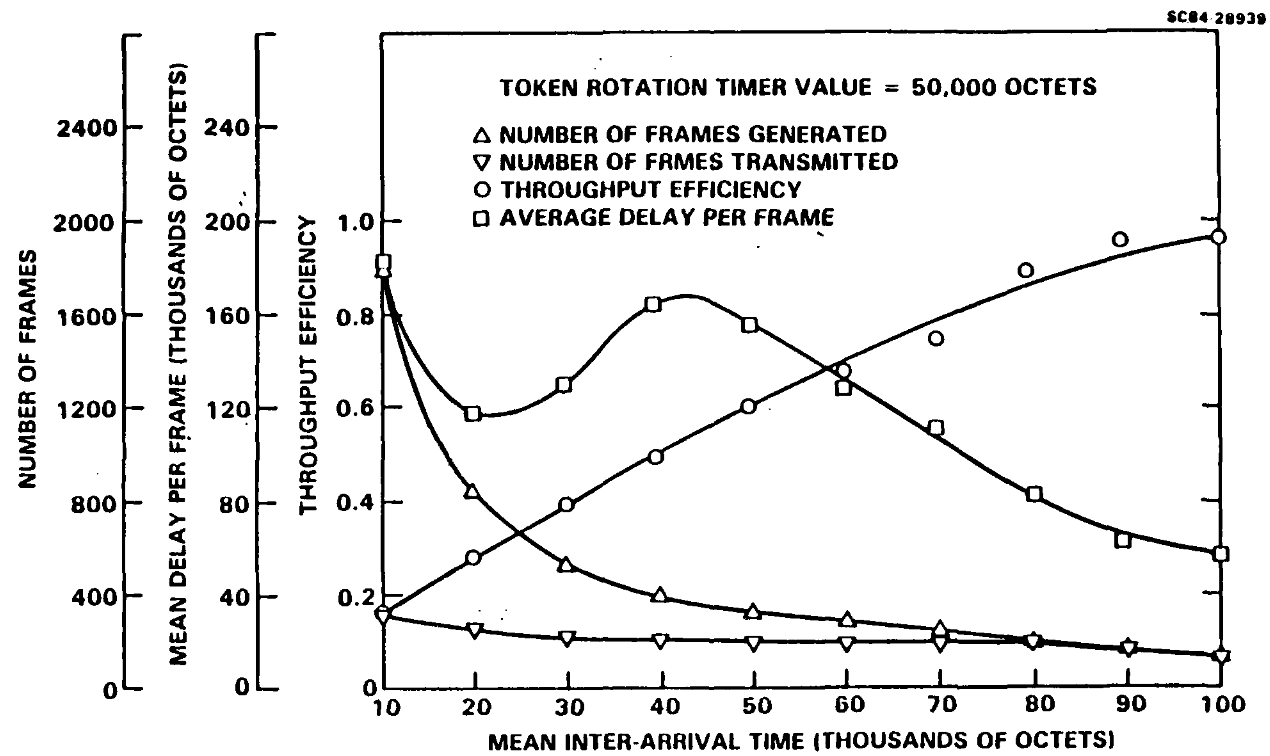
SLIDE 7


SC84-29206
HIGHER LAYERS
DATALINK LAYER
LOGICAL LINK CONTROL (LLC)
SUB-LAYER
DATA
FRAMES
MEDIUM ACCESS CONTROL SUB-LAYER
(TOKEN-PASSING BUS)
ACCESS CONTROL AND DATA EXCHANGE
1. TRANSMIT AND RECEIVE DATA FRAMES
2. TRANSMIT AND RECEIVE CONTROL FRAMES
3. CHECK PRIORITY CLASSES
4. MONITOR STATES OF THE TIMERS
5. ADMIT NEW STATIONS
6. RESOLVE CONTENTIONS
7. UPDATE LISTS OF ACTIVE STATIONS
8. DETECT DUPLICATE ADDRESSES
STATION MANAGEMENT
1. INITIALIZATION PROCEDURES
2. TIMER SETTINGS
3. CHANGE OF STATUS ON LOGICAL RING
4. GENERATION OF ADDRESSES
5. SPECIFICATION OF PRIORITIES
6. CHANGE OF PARAMETER VALUES
PHYSICAL LAYER
DATA AND CONTROL FRAMES

SLIDE 8

INFLUENCE OF MESSAGE PRIORITIES

- AVERAGE DELAY OVER ALL PRIORITY CLASSES IS DIFFICULT TO INTERPRET (Ex: TOKEN PASSING BUS)



SLIDE 9

Rockwell International

RUN TIME REDUCTION IN DISCRETE EVENT SIMULATIONS

- PURPOSE OF THE SIMULATOR – OFF LINE DESIGN AID OR REAL TIME AID FOR SYSTEM MODIFICATIONS
- APPLICATION LEVEL PROCESSES ARE SLOWER THAN UNDERLYING COMMUNICATION ENTITIES
- THIS INCREASES THE NUMBER OF EVENTS TO BE SIMULATED ENORMOUSLY

Solutions:

- WHEREVER POSSIBLE 'COMPUTE' THE IMPACT OF EVENTS (EX: IDLE ROTATIONS IN A TOKEN PASSING BUS NETWORK)
- DISTRIBUTED SIMULATIONS

SLIDE 10

Rockwell International
Science Center

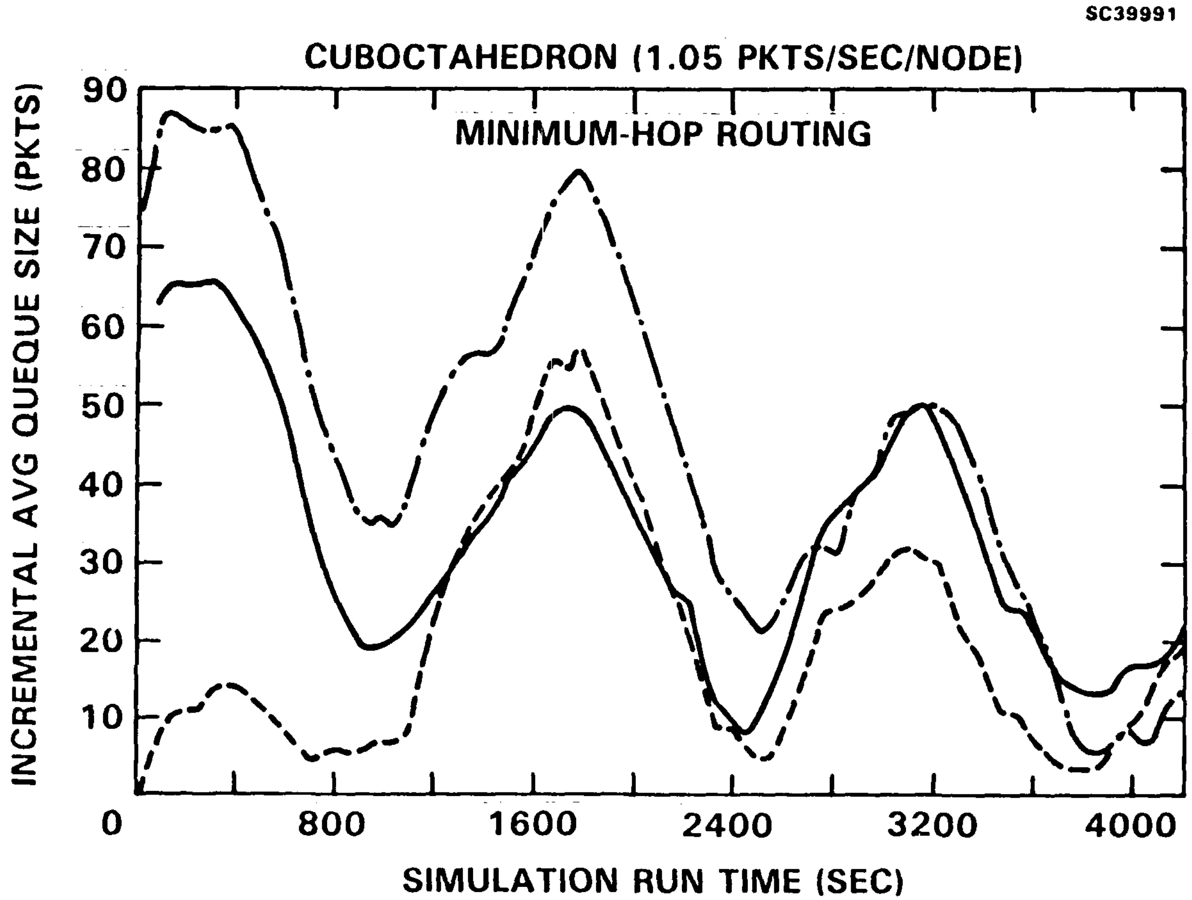
VALIDATION OF SIMULATION RESULTS

- VALIDATION OF STEADY STATE RESULTS – CONFIDENCE INTERVALS?
- CAN ALL NETWORKS HAVE 'STEADY STATE' SITUATIONS?
- HOW TO RECOGNIZE EXCEPTION CONDITIONS OR TRANSIENT SITUATIONS LEADING TO INSTABILITY?
- 'INITIALIZATION' OF NETWORK SIMULATORS – WITH EMPTY BUFFERS OR PRE-SPECIFIED QUEUE SIZES?

SLIDE 11

Rockwell International
Science Center

MULTIPLE SATELLITE NETWORK SIMULATION: MEAN QUEUE SIZE Vs TIME, VARIOUS INITIAL STATES

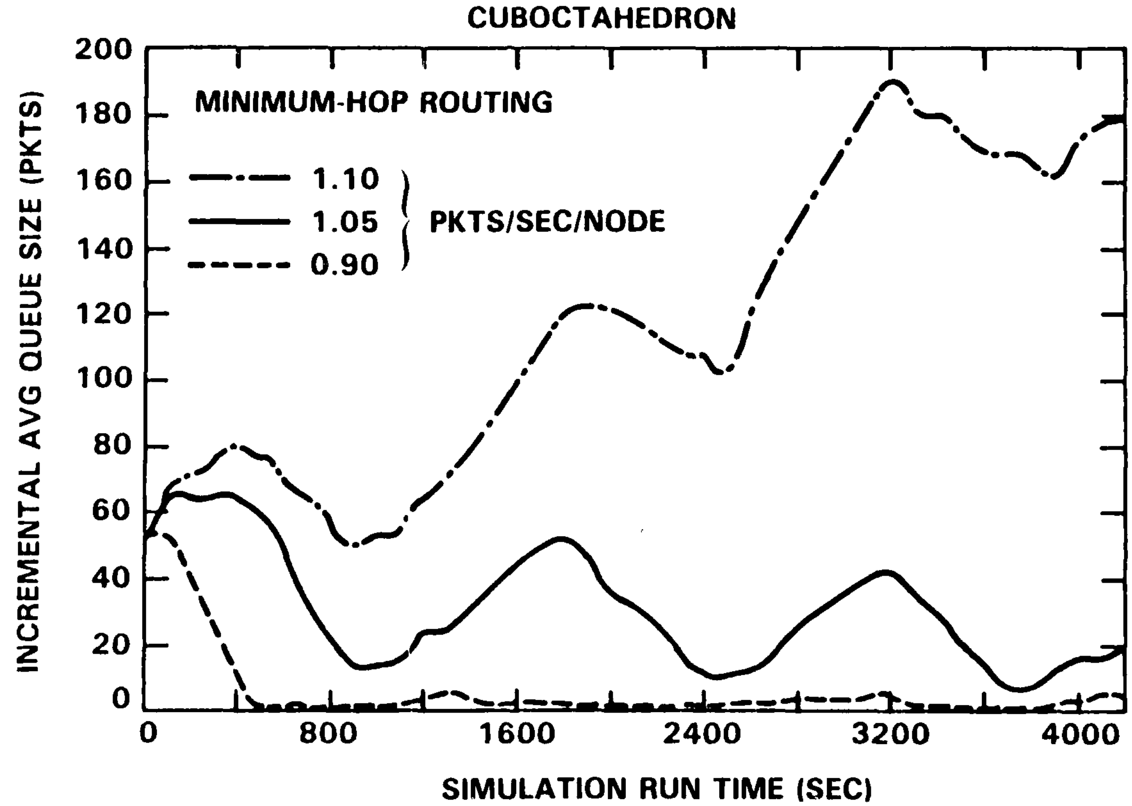


SLIDE 12

Rockwell International
Science Center

MULTIPLE SATELLITE NETWORK SIMULATION: MEAN QUEUE SIZE Vs TIME, VARIOUS OFFERED TRAFFIC LOADS

SC40392

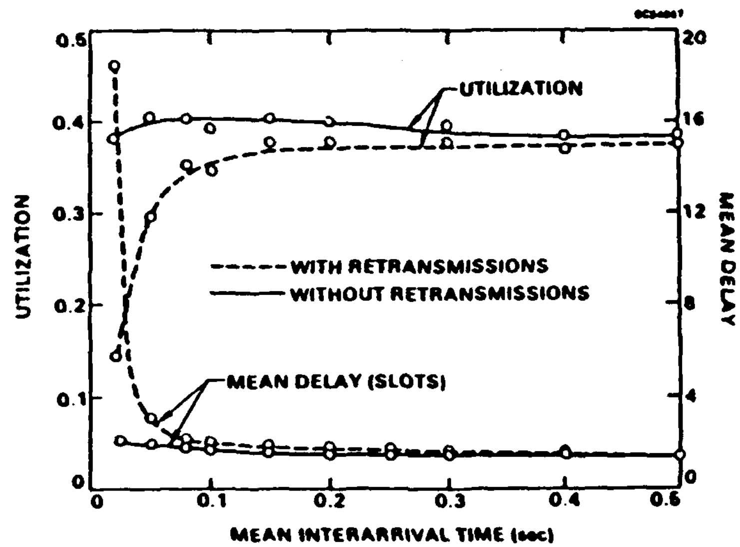


SLIDE 13

Rockwell International
Science Center

PERFORMANCE EVALUATION OF MULTI-HOP NETWORKS

- SIMULATIONS ARE PARTICULARLY USEFUL IN THIS CASE AS ANALYSES ARE VERY DIFFICULT



However

- WHAT ARE THE APPROPRIATE MEASURES OF PERFORMANCE?
- HOW TO DEFINE 'THROUGHPUT' OR 'UTILIZATION'?
- SHOULD AVERAGE MESSAGE DELAY INCLUDE ACKNOWLEDGMENT DELAY?
- SELECTION OF RIGHT PARAMETERS WILL FACILITATE MORE EFFECTIVE USE OF SIMULATION MODELS

SLIDE 14

Rockwell International
Science Center

SUMMARY

- **NETWORK SIMULATIONS CAN PROVIDE ADDITIONAL INSIGHT THAN OBTAINABLE THROUGH QUEUEING ANALYSIS** (*Ex: Priorities, Routing, Delay Distribution, Etc.*)
- **MORE REALISTIC TRAFFIC MODELS CAN BE CONSTRUCTED AND USED**
- **TRANSIENT BEHAVIOR CAN BE OBSERVED** (*Ex: Bottlenecks, Dependent Traffic, Etc.*)
- **INTERPRETATION AND GENERALIZATION OF RESULTS HAS PITFALLS**
- **ANALYTICAL TOOLS CAN AID SIMULATIONS** (*Ex: Methodology to Validate Results, Priority Handling Algorithms, Estimating Distribution Tails with Insufficient Samples, Etc.*)
- **DISTRIBUTED SIMULATION MAY BE USEFUL IN SPEEDING UP RUNS OF LARGE SIMULATION MODELS**

SLIDE 15

Rockwell International
Science Center

has push-button protocols in it and some graphics. Have you considered this?

SASTRY: We did consider that. We did not use it because as you saw earlier, we had so many different models built. Once you buy a readymade package, there is always a problem as to how to modify that for quite a different network. We always found it easier to start from SIMSCRIPT and quickly build models. We have 3 people who are full-time programmers and in fact as you keep talking to them, they talk back to you in the SIMSCRIPT code, that is the experience level. In this way we build prototype models very quickly. So we thought it was not going to be advantageous for us.

PICKHOLTZ: Is it therefore written in SIMSCRIPT?

SASTRY: Yes, it is written in SIMSCRIPT. Suppose you want to have a very wide variety of networks, like multiple satellite networks, for example, or a local area network; they are vastly different networks. Our interests are so wide, we thought we are better off starting from SIMSCRIPT and writing our own code, rather than using readymade packages. Incidentally, we are using the PC version of SIMSCRIPT, and most of the diagrams that you saw are all displayed on the PC. We have a highly effective user interface, thus by using a sort of a menu process, the user can put inputs and get displays, and PC SIMSCRIPT also has animation. It is through the animation that I was able to show you the transient results. And SIMSCRIPT is also coming CACI, in about 3-4 months on SUN. It is available on VAX 11-780s, that's where we developed our simulations. It is also available on IBM machines, so SIMSCRIPT is preferred as our simulation language.

SHANMUGAN: Our next speaker is Jim Kurose from the University of Massachusetts. Jim is going to talk about current CAD tools like BOSS at the network level.

KUROSE: I'd like to talk about CAD tools for networks and give an overview of what I see as state-of-the-art. I come from a Computer Science department as opposed to an Electrical Engineering department. The first thing that means is that I don't call them tools, I call them an environment which is what all the software engineering people call these things. At the heart of any kind of environment for doing any kind of modeling, I've got an underlying language. It might be a language like SIMSCRIPT, it might be a block-oriented language with the primitives defined by BOSS, it might be something like RESQ. So I can have any language, and presumably in the middle of that I have some model-database of models constructed within that language.

Now the reason why it's called an environment is because I have a set of tools which are all geared towards this particular language. I have tools for specifying my models, for building them, for solving them, analytic tools, simulation tools, output analysis tools for looking to the results, experimental design facilities and utilities, etc. The important point here is that all of these tools should see a common interface in terms of what a model looks like, and they should all be expressed within this language so I don't have a disparate group of tools that I'm trying to apply to models written in different languages. I would like to draw an analogy between what we are trying to do here and what people have already done in software development environments. I think that's an area in which computer scientists are generally quite far ahead. I think a number of other

people have made that observation.

The way I would like to structure my talk is to briefly give you an idea of what's going on specification tools, solution tools, output analysis tools, and a design facility. Before I do that I would like to talk about one common feature that I see in absolutely every network modeling package that people are working on, or environments that people are working on, and that's the role of graphics. I won't say a lot on that because Professor Turin provided a very eloquent description of why it's good. My view is that it provides a single, uniform interface to all aspects of the modeling process. I construct something graphically, I solve it graphically, I animate my solution, if I'm doing simulation. When I want to see my output results, just like in BOSS, I do that graphically. I point to it, I say "Look, *I'm interested in this queue*, let me see time-delay, let me see throughput utilization, etc." and finally, a point that is often overlooked (I always overlook it, but the people I work with in IBM never overlook it), is how important it is to communicate results to other people. If you walk in and say, "Here's my model, and here are my results", and you unfold 500 pages of computer paper, people aren't going to follow what you are trying to say. But we've found that graphical models and the ability to animate a solution and actually show the model running provides a very effective means for communicating, both the structure of a model and the results of the model to other people.

Here are some examples of systems that rely very heavily on graphics. The performance analysis workstation being done by Ben Melamed at AT&T, "RESQME" which is the Research Queueing Package Modeling Environment. I promised myself I wouldn't give a sales speech here, but let me just say that if anybody is interested, I would be happy to provide more details about what we are doing there with people from IBM. If you're from Massachusetts, you are welcome to come and visit and I would be happy to show you around. "PAWS" is being done by the University of Texas -- IRA, SIMSCRIPT, SIMVISION, SIMAN and CINEMA, there are a whole number of these. It is very clear there's an important trend there.

In terms of the underlying language, the goal of any environment is to define a very rich, but hopefully a minimal set of modeling primitives or building blocks which are appropriate to the modeling domain. Here are examples that are taken out of existing packages: A queue modeling element for instance, split and fission nodes where messages coming through can make copies or related copies of itself, a multi-access channel port, so when a message passes through I start simulating the propagation of the message on a channel, or, here's another one for an IBM SNA virtual circuit with pacing. So as you can see, there is a wide range of modeling elements that exist underlying a language, and in general there is an important tradeoff between flexibility and effort. If you are trying to build a model of SNA, it is very easy to do it in this, but very hard or time consuming to do it in this. However, if you are building your own routing algorithm, and you are interested in studying its performance. it is very hard to do it, as a matter of fact it is impossible to do it with this kind of modeling environment. It is much easier to do it here. One thing I see as very important for all modeling environments is language support for structuring very large models. I think if there is anything that is going to stop these modeling environments dead in their tracks, it's the

problem of what happens when networks get very big, and I try to model such big networks.

Another potential problem is that most of these environments are based on 1960s or 1970s packet switching technology. Well, what about high-speed networks built around interconnection networks, what about ISDN, what about packet radio networks?

What I'd like to do now is to just go through very quickly: model constructions, model solutions and output analysis. In terms of the construction, again Professor Turin discussed it somewhat this morning. When I build my model, I simply choose icons off of a palette and put them on a modeling area. All modeling elements have certain attributes associated with them. For instance a queue has a name, it has service time of messages going through, if it's a priority discipline, it has priority information, if it's pre-emptive priority, it has preemptive priority information. This is typically specified in language specific context sensitive forms: we can learn a lot of good things from compiler technology, such as incremental parsing, so I can have immediate error detection. As soon as I make a mistake, a red light goes off, the icon turns off and I see, "Gee, I've made a mistake." I don't wait till I've built the model and then get 500 million error messages coming out. The goal is to make it very hard if not impossible to build an incorrect model. In terms of support for large models, panning and zooming over the modeling area is the typical feature.

Secondly, I can have graphical support for model structuring so I can build hierarchical models. Here's one modeling plane, if you will. I have a high level model here, and I've got a black box here, and if I'm working on this graphically, I press that black box and all of a sudden I'm down here, and I'm working down here in this modeling plane. We've found that this kind of hierarchical structuring has been about the most successful thing we have been able to come up with in solving large models.

(Skip those two slides...) O.K. Model solutions: What are people doing there? Well, there's the old question, should I do it analytically, or should I do it through simulation? Is the complexity of the networks that we are building such that analysis is really not an option? I'll just tell you a statistic within IBM. Over 99% of all the models are being solved through simulation even though we have state-of-the-art queueing theoretic algorithms for solving queueing networks. Some people are still solving it analytically and another possibility is hybrid techniques. We've seen a number of people talk about the animation of a simulation solution so I won't talk about that. But we've also had some people mention distributed parallel simulations and I would like to talk about that.

The question here is, I have two options. Should I take a single simulation and distribute it among all the processors that I have, or should I take multiple simulations and give one simulation to each processor and have each processor work on the simulations and gather all the results from all the simulations and make some inferences. Well, the most recent results I've seen are sort of bad news for both of these techniques.

In terms of distributed simulations (one simulation over an entire group of processors), there was a paper in *Sigmetrics* about three weeks ago that did some measurements on a Sequent machine. They were

looking at a central server model and they were simulating it using a distributed simulation algorithm running on 5 processors. They found that there was a difference between a factor of 16 in doing it on a single processor versus 5 processors. Well, you think, certainly it can't be a factor of 16 speed-up with 5 processors. In fact, 5 processors were 16 times slower than a single processor. So I think that's indicative of some of the problems involved there.

In terms of parallel simulations, you have to be very careful with what you do. If I gave each of you simulations to run and looked at my watch and stopped you after 10 minutes and gathered all the results and looked at mean, or some kind of estimates, unless I was very careful about which of your simulations I used and which I threw out, I'd get biased results for what I was trying to look at. So, again, I think both in parallel and distributed simulations people have been saying, "This is the way to solve our computational problems." I'd like to say I think there are some interesting and important issues there that have to be addressed first.

In terms of output analysis, if I have a graphical model representation, then a graphical display of results is the obvious thing to do. A shortcoming I see in many tools is that you run a simulation and get some performance results, but only point values. I think people here are sophisticated enough to realize that if I have a point value without confidence interval that can be pretty meaningless. So it is important to provide multiple confidence interval techniques, generative methods, batch means, independent replication.

There has been a lot of talk about A.I. I wasn't sure I was going to put this up, but people who think about A.I., usually fall in one of two camps. Either you love it, or you hate it. And I suppose I would fall in the nattering nabobs of negativism, I think that's the phrase. However, I do think this is one possible application for AI systems. Because when you're trying to select a confidence interval technique, that is a very narrow area of expertise in terms of selecting the methods for generating confidence intervals, for projecting how confidence intervals will change as a function of how long I do simulations for choosing run-length. I think that is a rich area to be looked at.

So we've looked at how to build a model and how to solve a model and looked at the output results. If you remember from the beginning, I talked about an experimental design facility. This is something I have been working on with some people at IBM. The idea here is when I perform a simulation it's usually in the context of some kind of parametric study of system performance. The way it's usually done is, I set some parameters, I simulate it, I get some results, I tweak some parameters, I get some more results. If the results were better, I tweak the parameters in that direction again. Now, why can't I automate that, why can't I optimize the performance of my model somehow over a set of tunable parameters?

There are some interesting applications here of stochastic approximation algorithms, and then there is the question of how I should do this optimization. Should I do it on central differences like in Kieffer-Wolfowitz that turns out to be very slow? Could I come up with some kind of approach for estimating gradients in just one simulation run? There is a technique that some people have come up with called perturbation analysis, that's one possibility, but there might be some convergence prob-

lems with that. There is a technique recently developed by some people at AT&T Bell Labs based on likelihood ratios for estimating these gradients. So there are number of open questions there, too.

When I tried to tell the people at IBM that this was an interesting problem to look at, basically because I found it to be a mathematically interesting problem, that I could probably prove some convergence results and things like that, they said "Well, you are not asking the right question here. You are an academic and you are not worried about cost of the systems and things like that." You really don't want to optimize the performance of a fixed resource system. The most important question that people in the "real" world want to answer is, "How can I find the minimum configuration meeting the performance requirements, or similarly, how can I maximize the performance over configurations meeting some kind of fixed cost?" These are problems that we are going to begin looking into.

This is my last slide. What are the challenges? One of the first challenges I see is that the development of any kind of environment is necessarily evolutionary. I have been working with these people from IBM for 8 or 9 years now. We have gotten lots of feedback along the way from users in terms of how to modify the underlying language to make it rich enough so that it is desirable to a broad set of users. I think these new graphic requirements haven't had that time to get that kind of feedback yet. In terms of other challenges, new network architectures in switching technologies, particularly for high speed networks, might require rethinking the modeling primitives and solution techniques that we use. There is an obvious overriding concern as far as I can see it with what happens when my model gets very big. Any picture anybody will ever show you, of any graphical environment will have a very small model. Sam had a picture of 4 icons up there. Well what if it's 400? I had a bunch of pictures I skipped over and I had 8 icons up there. Maybe we are halfway there towards solving the problem as opposed to Sam, but I think that's a really serious problem.

Finally, a problem that Dr. Sastry mentioned with high-speed networks, there is an impending significant change in the time-scale. I'm talking about switching things at gigabit rates in large switches and if I'm going to try and model individual packets or even messages going through, that's just going to eat up too much simulation time. So that's another problem that will have to be addressed.

SHANMUGAN: Thank you Jim for an excellent presentation and also for staying on time. Any questions for Jim?

PURSLEY: Your observation about the 1970s technology for the networks, in what's available, is there anything improving there? For example, in packet radio networks? What's available for packet radio networks that take into account the realistic communication environment for such networks?

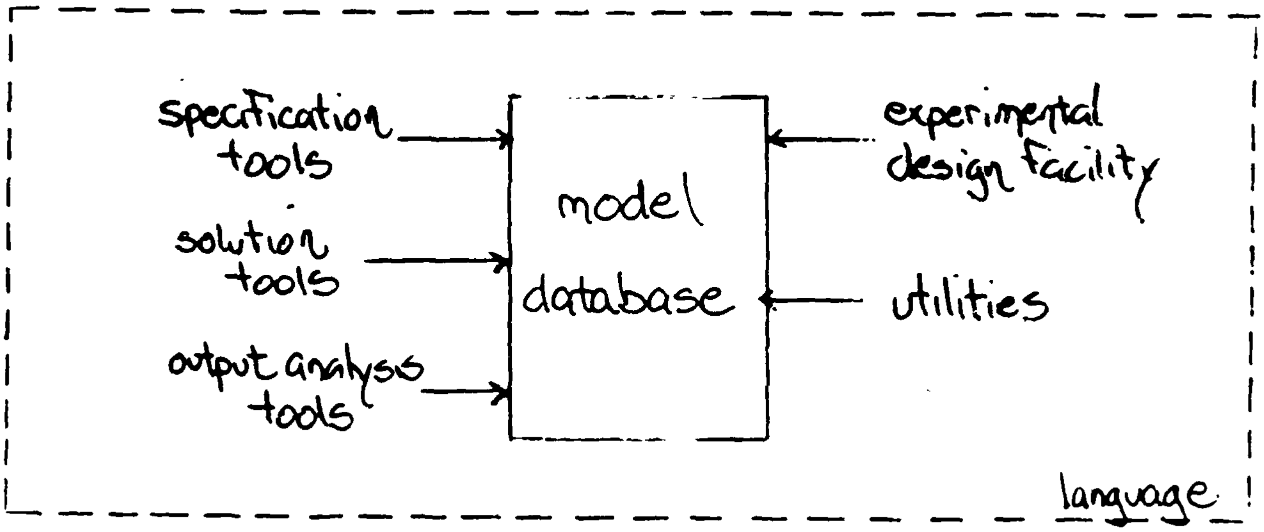
KUROSE: I had a paper in the last ICC that talks about a modeling environment specifically geared towards multiple-access communication systems, because the simultaneous possession of the channel and the problems of actually modeling the propagation delay in a single channel turn out to be very hard within a conventional queueing theoretic paradigm.

SASTRY: Regarding the packet radio network we applied shortest path type routing, and then establishing good neighbors, we did not go through that. Most of the DARPA packet radio protocols are based

CAAD Tools for Networks
"State of the Art"

Prof. Jim Kurose
Department of Computer and Information Science
University of Massachusetts
Amherst, Mass. 01003

An Integrated CAAD Environment



- underlying high-level language or set of modeling primitives

- set of tools for constructing, solving, analyzing, manipulating models and results

- analogy to software development environments

SLIDE 1

The Role of Graphics

- graphical representation most natural form for expressing network model.

- single, uniform interface to all aspects of modeling process:
  - construction
  - solution/animation
  - output analysis
  - communicating results to others

- visual programming paradigm

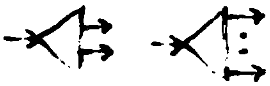
- examples: PAW(AT&T), RESQME(IBM/UMass) PAWS(Texas/IRA), SIMSCRIPT/SIMVISION, SIMAN/CINEMA

SLIDE 2

The Underlying Language

- GOAL: set of modeling primitives (building blocks) appropriate for modeling domain.


a queue


fission/split nodes


multiaccess channel port

IBM/SNA VC with pacing

- tradeoff: flexibility versus modeler effort

- language support for structuring large models

- most languages based on 1970's packet switching technology. What about HSN, ISDN, PRnet, ...?

SLIDE 3

Model Construction

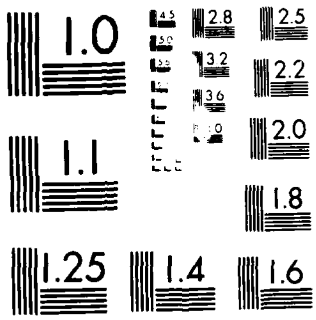
- graphical editor:
  - icon selection for palette(s)
  - attribute specification in language-specific context-sensitive forms
  - incremental parsing, immediate error detection

- support for large models:
  - panning, zooming over modeling area
  - graphical support for model structuring

SLIDE 4

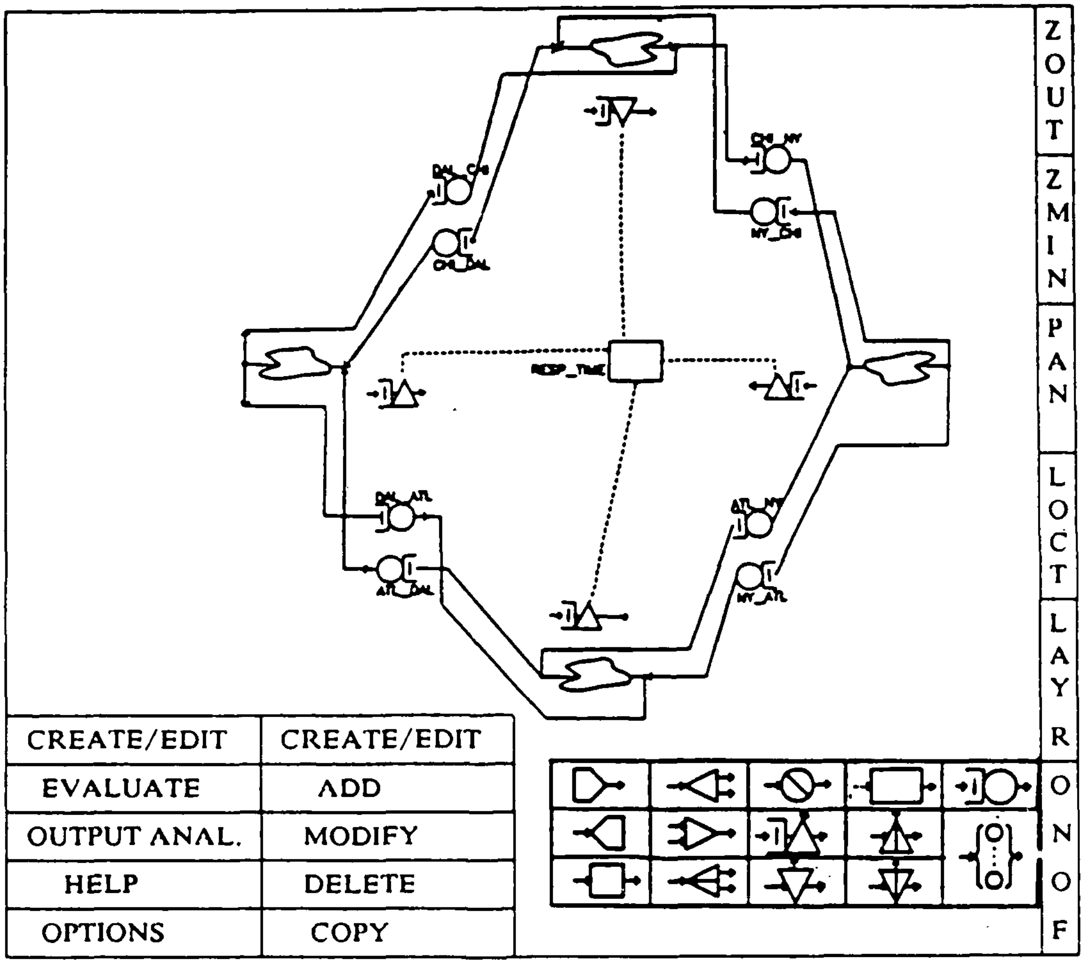
AD-A194 336 ADVANCED COMMUNICATION SYSTEM ENGINEERING PROCEEDINGS 3/5
WORKSHOP HELD IN SE.. (U) UNIVERSITY OF SOUTHERN
CALIFORNIA LOS ANGELES COMMUNICATION S.. R A SCHOLTZ

UNCLASSIFIED MAR 88 CSI-88-03-02 ARO-24913.1-EL-CF F/G 25/5 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

Example: RESQME



SLIDE 5

Example: RESQME

```
RESQ Subsystem              Model: FOURNODE  03/10/86 4:18p
CREATE/EDIT                  Submodel:
=====================================================
QUEUE: Chi__NY__q
  TYPE: prty
  CLASS LIST: Chi__NY
    SERVICE TIMES: constant(jv(msg__leng)/9600+prop__delay)
    PRIORITES: jv(msg__type)




================ Expected Action Summary ================
Enter priority level for this class (smaller number implies higher priority)
================== Message Window ==================



1Help 2Select 3Duplic 4Delete 5Insert 6Up 7Down 8Top 9Bottom 0Return
```

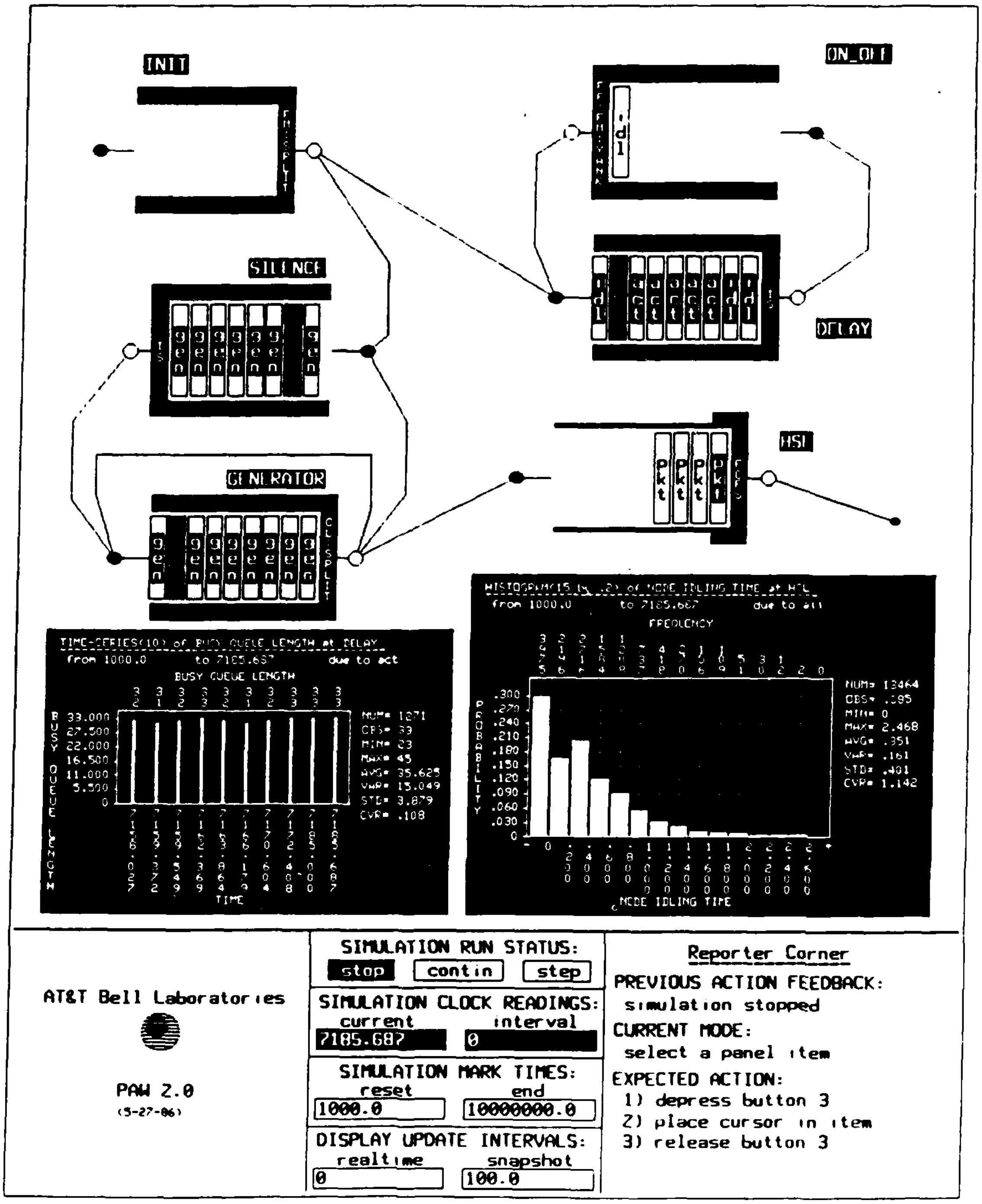
SLIDE 6

Model Solution

- analysis versus simulation
  - will network complexity exceed analytic capabilities?
  - hybrid techniques
- animation of simulation solution
  - job flow and time varying statistics
  - visual demonstration of interrelationships in model structure
  - model debugging
  - communication of model and results to others
- distributed or parallel simulations:
  - single distributed simulation on many machines has high synchronization / communication overhead
  - many parallel simulations: unbiased estimates may mean less than $O(n)$ speedup.

SLIDE 7

Example: PAW



SLIDE 8

Output Analysis

- graphical model representation $\leftrightarrow$ graphical display of results

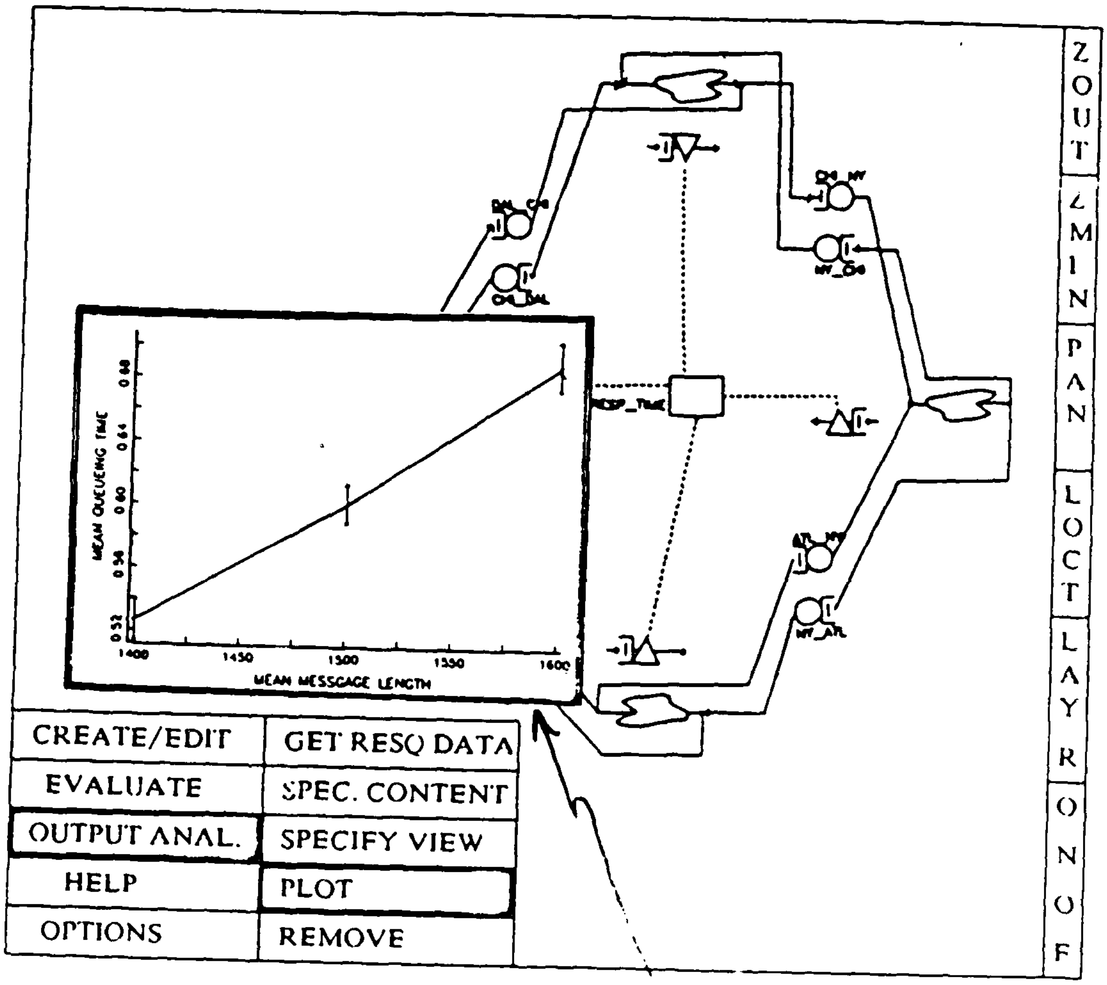
- statistical analysis of a simulation
  - many CAAD tools provide only point estimates of perf. measures!

  - multiple confidence interval (CI) techniques should be provided: regenerative, batch means, ind. replications, ...

  - applications for an "expert advisor":
    - selecting CI methods
    - CI projections
    - choosing run lengths

SLIDE 9

Example: RESQME



SLIDE 10

display results by pointing to icon

Experimental Design Facility

- automated parametric optimization
  - optimize performance over set of tunable parameters
  - optimization based on: central differences (slow?) or gradients (how to estimate?) of performance measures.

- are we asking the right question?
  - optimize performance of fixed resource system?
  - find minimum configuration meeting perf. requirements?
  - find maximum perf. configuration meeting fixed cost?

SLIDE 11

The Challenges

- development of integrated network CAAD system is evolutionary
    - tools now under construction
    - relatively little feedback thus far

- new network architectures and switching technologies require rethinking modeling primitives and solution techniques.

- problems with scaling size of model

- impending significant change in time scale of networks.

SLIDE 12

on that, we have a multi-hop packet radio simulation model but we want to still further work on that with other access mechanisms other than CSMA type in the future.

KUROSE: But I'd also like to draw the distinction between a specific piece of software to model a specific network in a specific set of protocols. For instance, I know at SRI, Nachum Shakham's group has lots of simulation models that they've built for looking at multi-hop packet radio networks, but it is not a general purpose tool so that if you want to configure your own network, to look at your own algorithms, it cannot provide the building blocks. It's a much more specific application, and a specific design to a packet radio network.

SHANMUGAN: Thank you Jim. Our last speaker is Dr. Hussein Mouftah from Queens University. He is going to wrap up the session by giving us a brief overview of the role of expert systems in CAD tools at the network level.

MOUFTAH. First of all, I should ask you to excuse me if you find me a little bit exhausted from the fact that I have just arrived this morning from Boston, and my flight was at 6 a.m. from Boston, and I have been up since 4 a.m., which means about 1 a.m. here.

My talk in the next few minutes will be on expert systems for modeling analysis and design of communication systems. Just a working statement, as Phil was mentioning, it is open for discussions. I will also try to answer the question that came from here, as to what is an expert system. I will start with a short introduction, followed by a specific set of objectives, and then I'll talk about the different types of performance evaluation tools, analytical modeling and simulation in real time measurements, including also the perturbation analysis techniques and then, specific emphasis will be put on the network's performance advisor, followed by some conclusions. I am calling these conclusions, because I alluded to an introduction in the beginning. I should say, however, that it is a summary that will also point out topics for research work in this area.

The main objective here, of course, is to provide tools to study the performance of communication networks. The performance evaluation tools can be done or carried out using different techniques. For example, at the beginning (VIEWGRAPH M4) we have the analytical modeling, and I can characterize it by the fact that it can give us an exact solution for simple models. However, it would give us an approximate solution for more complex models. As the model becomes more complex, then we have to make some assumptions that will reduce the validity of the results. There would be approximate results. Analytical modeling is usually cost effective, because of the simplicity of the model and less time consuming because it is a straight forward approach.

Simulation techniques are useful for complex systems, and this is when the analytical models or analytical solutions would fail because of the severe assumptions that we would have to make at that time. So, simulation is useful for complex systems, and closer to real systems, however requires extensive knowledge of the simulation language. So we would then have to familiarize ourselves with these simulation packages before we use them. That's a consideration we should keep in mind, unless it is as user-friendly as BOSS. However, sometimes they are expensive especially if we are not using the right

language for our application, then it would be time consuming and then again these two things (VIEWGRAPH M4) could come together and eventually be expensive.

The third technique would be using real time measurements from the network itself; some equipment would be used to measure the performance of the network running in real time, which achieved a high level of accuracy, but it can be used only if the real time systems are available, which is not usually the case. At least not in the design and development phase, so we cannot use it before the network is there. Also performance test drivers that can load the traffic and the specific desired data traffic characteristics, would not usually be available. These are some limitations on using real time systems.

Finally, there is the possibility of having some sort of combination of simulations and analytical modeling or real time measurements and that would be the case of perturbation analysis. This is less expensive than simulation. We can have some sort of expectations based on just the output of a single run, or we can also perform perturbation analysis based on some real time measurement data that come up from real time measurements. Useful for real time applications but accurate only for small perturbations, I'm going to say something about it near the end.

As we have seen all these techniques have some advantages and some disadvantages. So there is a need to decide when we use this and when we use the other technique. What we propose here is the use of a network performance advisor, which is a knowledge-based system that is also called expert system.

But what is an expert system, I'll try to answer the question for Frank. An expert system is a system which simply speaking mimics an expert on the computer. We simulate the expert on the computer. What is a simulation? Simulation is a program that can describe the event as it happens on the computer and then follow the sequence of events, apply some inputs, and provide the results at the output. We are simulating the system, or the device, or the event as it happens. Now what is an expert system? In an expert system, we are simulating the expert himself, the person who is an expert in a specific field. We are simulating him on the computer. Simulating him means to simulate how he thinks. That's where artificial intelligence comes into play We are simulating how he thinks by having some blocks of software part of our program go through a number of effects that are describing the system that we are studying in a certain way. And this certain way is defined by the expert through that module. So these are the three different modules that form an expert system (VIEWGRAPH M5). So again, the expert system is a piece of software that is composed essentially of three blocks, the inference engine, the knowledge base and the user interface, which communicates with the user and also communicates with the expert person who would load or prepare that type of database that I referred to call it here as knowledge base, as this type of database is defined by the people who specialize in developing expert systems.

The inference engine is the piece of software that would have the different techniques to go through a number of facts and rules that describe the system, and then come up with the results, with some advice to the user through the user interface. Now these three modules are summarized (VIEWGRAPH M6) here very briefly as follows.

The user interface extracts information necessary for processing the intended structure, and provides expert advice, and then communicates with the knowledge expert, the person that you are trying to mimic. So he is loading or preparing his knowledge-base through that user interface. He communicates with the system through that user interface, to gain additional knowledge by updating or modifying that knowledge-base. Now the inference engine is responsible for making the decisions by searching for relevant information from its knowledge-base, from this module (on VIEWGRAPH M6).

The decision is based on heuristic rules, or basic proposition calculus mechanism of inference and proof. The knowledge-base consists of a database of information, necessary for providing an expert advice and a set of assertions that relate the information pieces together. These definitions of these three modules are general, and they can be the elements of an expert system for any type of application, in medicine, in communications, for networks, etc.

Now in my talk here, for the rest of my 5 minutes, I will be talking about the application of expert systems as a tool for studying the performance evaluation of networks. What we can see in red (upper 3 boxes of VIEWGRAPH M6) is the elements of the general expert system, and what we can see in green (3 left-hand side boxes of VIEWGRAPH M6) is the simulation approach that the expert system can choose, and then what we can see in blue (2 right-hand side boxes of VIEWGRAPH M6), is the analytical modeling approach that the expert system can choose, and what we have here in the middle is the perturbation analysis that can be considered as a combination of more than one technique, as well as we can take the data from the simulation, or can take some data from real time measurement. So we have here an expert system that would define or choose for us the approach that we can apply and also, how we can apply that approach. Or in other words, how we can apply the simulation or the analytical model or the perturbation analysis. For example for simulation techniques, in case we choose that approach, what we have here is the simulation model library, then the simulation engine and a verification and validation processor.

Here we have a set of functional blocks that we can put together in order to model any sort of communication network, and then in the inter-simulation engine, we actually execute the model that we have chosen here or gathered or put together from the simulation model library, and apply our sequence of events as I described in the beginning, and then the verification and validation processor would analyze for us the output of that simulation engine and validate the results as well as decide precisely, with the help of the expert system here, as of when we can stop simulations and the confidence intervals that you heard Jim talking about a few minutes ago. If we need to change parameters and topologies, we can go back here (bottom left corner of VIEWGRAPH M6), alter them and run the simulations again. We can choose the analytical approach and put together or choose a specific model, an analytical model, that has been defined ahead of time.

As an example analytical models for networks are based on queueing theory, and they are usually in the following form (VIEWGRAPH M8). They would define for us the type of the process we have, its interarrival times, single or bulk arrivals, single or multiple servers, finite or infinite

EXPERT SYSTEMS

FOR

MODELLING, ANALYSIS AND DESIGN

OF

COMMUNICATION NETWORKS

- A WORKING STATEMENT -

BY

H.T. MOUFTAH
QUEEN'S UNIVERSITY
KINGSTON, ONTARIO
CANADA.

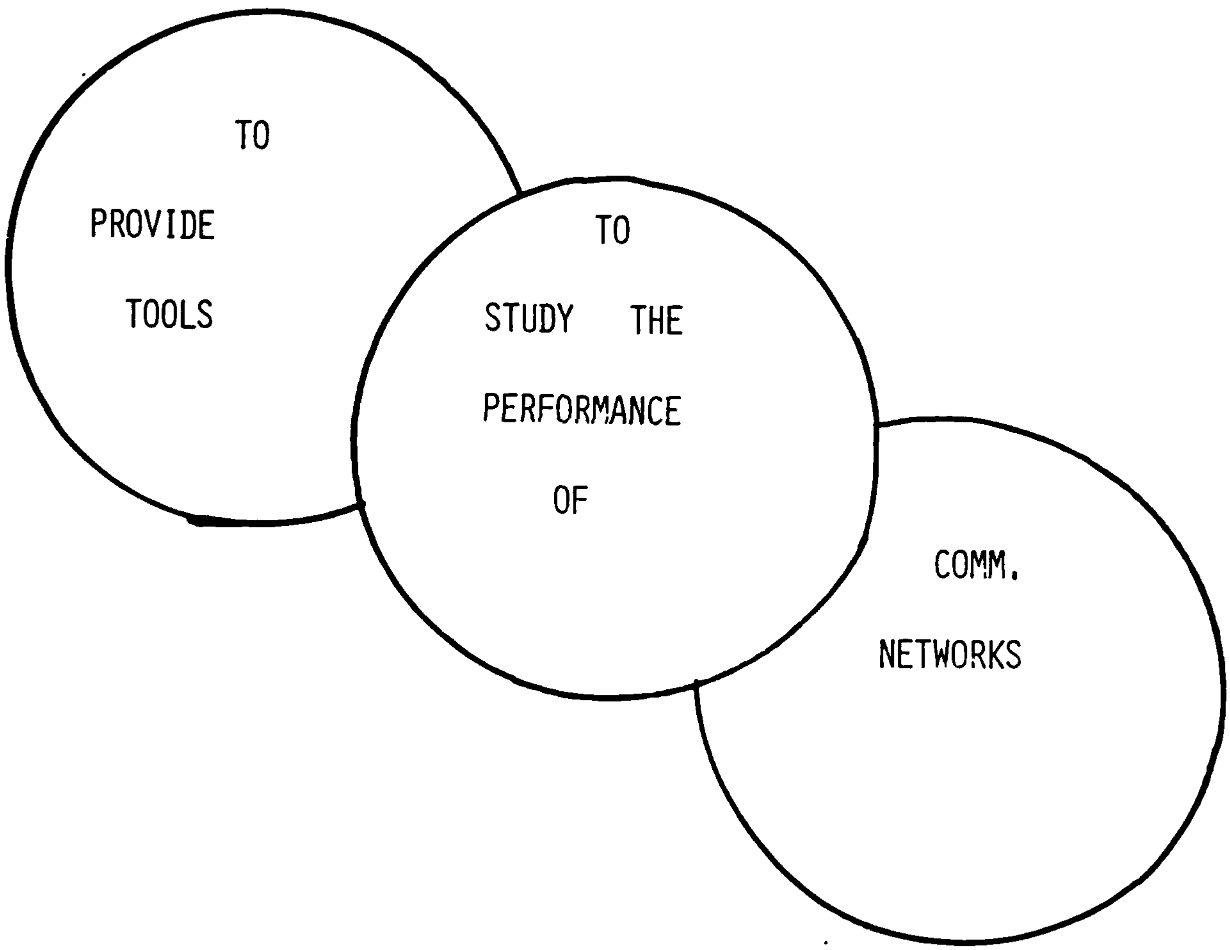
VIEWGRAPH M1

OUTLINE

* INTRODUCTION
* OBJECTIVES
* PERFORMANCE EVALUATION TOOLS
    - ANALYTICAL MODELLING
    - SIMULATION
    - REAL TIME MEASUREMENTS
    - PERTURBATION ANALYSIS
* NETWORK PERFORMANCE ADVISOR
* CONCLUSIONS

VIEWGRAPH M2

OBJECTIVES

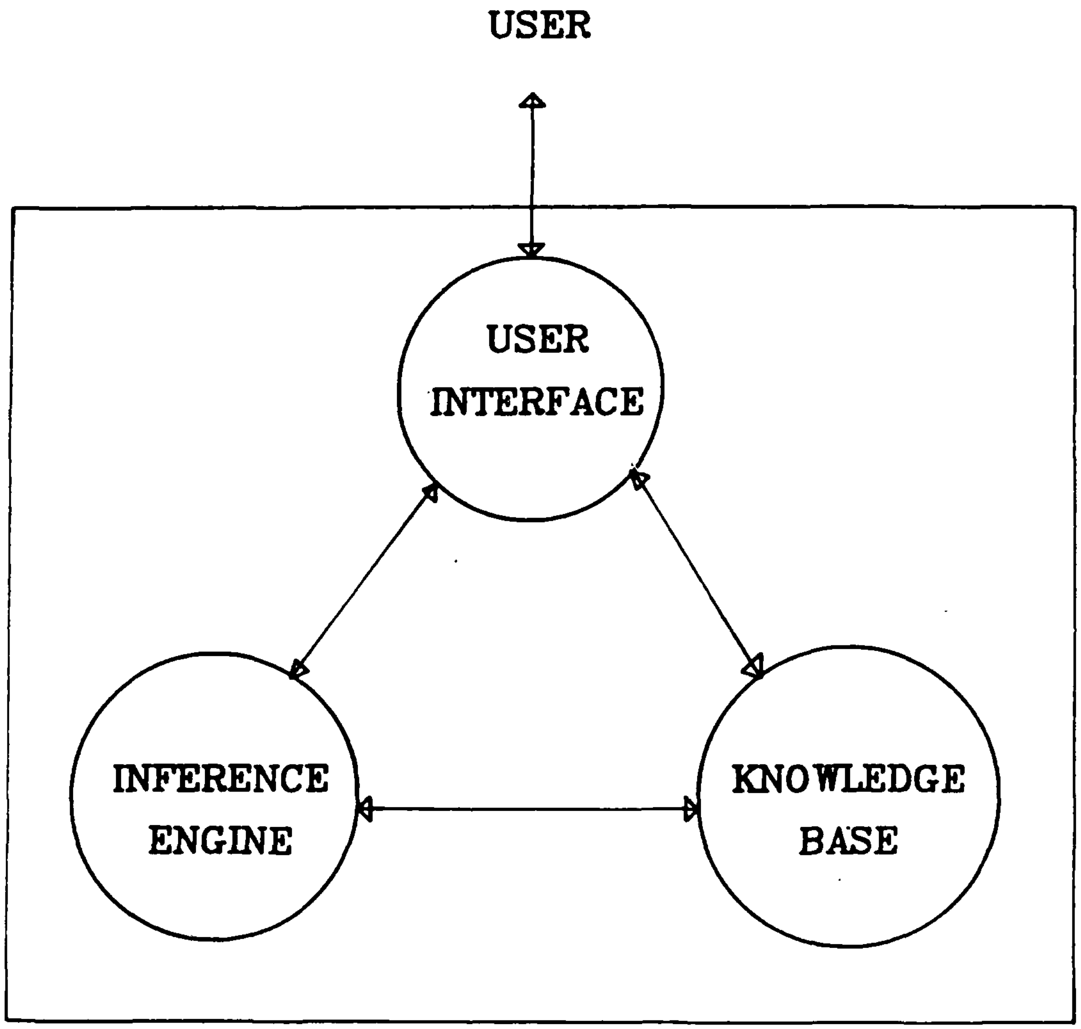


VIEWGRAPH M3

PERFORMANCE EVALUATION TOOLS

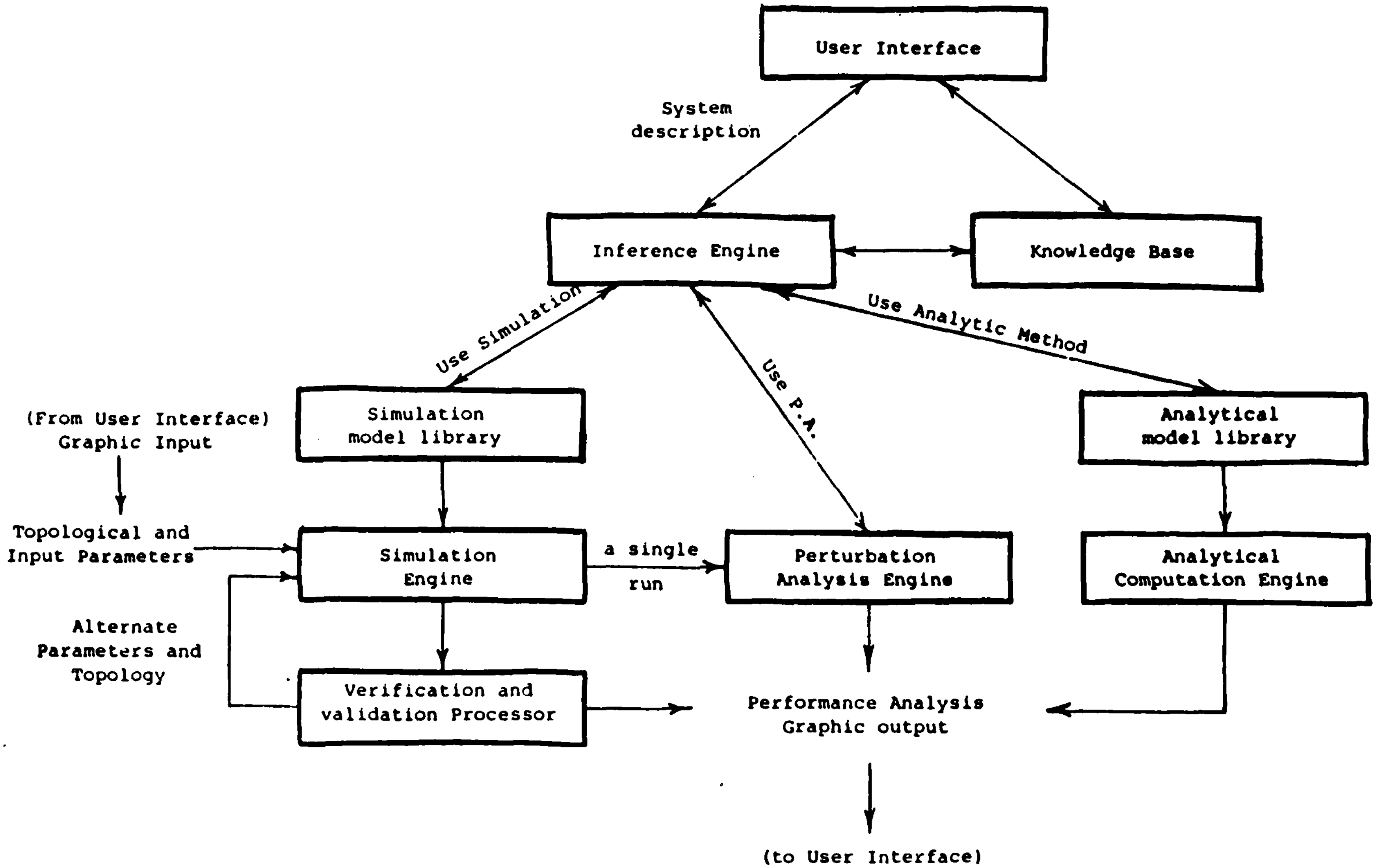
- ANALYTICAL MODELING:
  - EXACT FOR SIMPLE MODELS
    BUT APPROXIMATE FOR MORE COMPLEX MODELS
  - COST EFFECTIVE
  - LESS TIME CONSUMING
- SIMULATION:
  - USEFUL FOR COMPLEX SYSTEMS
  - CLOSER TO REAL SYSTEMS
  - REQUIRE EXTENSIVE KNOWLEDGE TO SIMULATION LANGUAGES
  - EXPENSIVE
  - TIME CONSUMING
- REAL TIME MEASUREMENTS:
  - ACCURATE, IF REAL TIME SYSTEMS ARE AVAILABLE
  - USUALLY NOT IN DESIGN AND DEVELOPMENT PHASE
  - USUALLY PERFORMANCE TEST DRIVERS THAT CAN LOAD THE SYSTEM WITH DESIRED DATA TRAFFIC CHARACTERISTICS, ARE NOT AVAILABLE
- PERTURBATION ANALYSIS:
  - LESS EXPENSIVE THAN SIMULATION
  - USEFUL FOR REAL TIME APPLICATIONS
  - ACCURATE ONLY FOR SMALL PERTURBATIONS

VIEWGRAPH M4



A Typical Expert System

VIEWGRAPH M5



Communication System Performance Advisor Flow Diagram.

VIEWGRAPH M6

THE NETWORK PERFORMANCE ADVISOR

* USER INTERFACE:

  - EXTRACTS INFORMATION NECESSARY FOR PROCESSING THE INTENDED FUNCTIONS AND PROVIDE EXPERT ADVICE TO THE END USER.

  - COMMUNICATES WITH A KNOWLEDGE EXPERTS TO GAIN ADDITIONAL KNOWLEDGE FOR UPDATING/MODIFYING THE KNOWLEDGE BASE.

* INFERENCE ENGINE:

  RESPONSIBLE FOR MAKING DECISIONS BY SEARCHING FOR RELEVANT INFORMATION FROM ITS KNOWLEDGE BASE. THE DECISION IS BASED ON HEURISTIC RULES AND BASIC PROPOSITIONAL CALCULUS MECHANISMS OF INFERENCE AND PROOF.

* KNOWLEDGE BASE:

  CONSISTS OF A DATA BASE OF INFORMATION NECESSARY FOR PROVIDING AN EXPERT ADVICE AND A SET OF ASSERTIONS THAT RELATE THE INFORMATION PIECES TOGETHER.

VIEWGRAPH M7

THE USER INTERFACE QUERIES THE FOLLOWING TWO TYPES OF SYSTEM INFORMATION:

- SYSTEM CONFIGURATION:
  - NUMBER OF QUEUES
  - SIZE OF EACH QUEUE
  - PACKETS FLOW ROUTES
  - ROUTING PROBABILITY MATRIX
  - SINGLE/MULTIPLE CUSTOMER TYPES
  - CUSTOMER SERVICE PRIORITIES
  - SINGLE/MULTIPLE SERVER SYSTEM

SUGGESTIONS FOR SELECTING APPROPRIATE OPTIONS

- TRAFFIC PARAMETERS:
  - ARRIVAL AND SERVICE PROCESS DISTRIBUTIONS
  - MEAN AND VARIANCE OF CUSTOMERS ARRIVAL/SERVICE
  - TYPE OF CUSTOMERS
  - NUMBER OF SOURCES
  - SINGLE OR BULK ARRIVALS

VIEWGRAPH M8

* ANALYTICAL MODEL LIBRARY:

CONTAINS SEVERAL KNOWN ANALYTICAL MODELS FOR EXACT SOLUTIONS OR WITH MINIMUM REALISTIC ASSUMPTIONS. (E.G. A(x)/B/N/K TYPE OF QUEUEING MODELS FOR MARKOVIAN OR GENERAL INTERARRIVAL AND SERVICE TIMES, SINGLE AND BULK ARRIVALS, SINGLE OR MULTIPLE SERVERS, FINITE OR INFINITE WAITING ROOM, AND MULTIPRIORITY SERVICE).

* ANALYTICAL COMPUTATION ENGINE:

CARRIES OUT CALCULATIONS ANALYTICAL MODELS SELECTED FROM ANALYTICAL MODEL LIBRARY.

* SIMULATION MODEL LIBRARY:

CONTAINS LARGE NUMBER OF MODELS OF VARIOUS FUNCTIONAL BLOCKS.

* SIMULATION ENGINE:

SELECTS A SET OF MODELS OF FUNCTIONAL BLOCKS AND CONNECTS THEM IN THE DESIRED TOPOLOGY.

VIEWGRAPH M9

CONCLUSIONS

* MODERN COMPUTER NETWORKS COMPLEXITY REQUIRES EXTENSIVE USE OF CAMAD TOOLS

* EFFORTS ARE DIRECTED TOWARDS DEVELOPING STANDARD CAMAD TOOLS FOR COMPUTER NETWORKS TO BE WIDELY USED (AS SPICE)

* KNOWLEDGE-BASED SYSTEMS WILL PLAY AN IMPORTANT ROLE IN DEVELOPING CAMAD TOOLS FOR COMPUTER NETWORKS.

VIEWGRAPH M10

buffers, as well as multi-priority service and so on. Then of course there is the other block we have here (right-hand side of VIEWGRAPH M6). It is the analytical computational engine that would carry out the calculations for the analytical models selected from the analytical model library. And I have already talked about the simulation model library and the simulation engine.

Now if we choose the third approach which is this one (center of VIEWGRAPH M6), the perturbation analysis technique, then here in this module, we estimate the gradient of a performance measure that we choose from a single run with respect to some input parameters and then the gradient estimates can be used in a subsequent analysis or optimization procedure. Again there are some limitations on the use of this analysis, which come essentially from the fact that perturbations should not be large. They should be sufficiently small so that the order of events in the realizations remains unchanged; otherwise, it won't give a good result. The inference engine would search for that PA if the PA can be applied or not in that particular use or application.

As I said in the beginning, for every introduction, I should have a conclusion, and here it is (VIEWGRAPH M10). As a conclusion I would say that modern computer network complexibility requires extensive use of CAD or CAMAD, for computer aided modeling analysis and design tools, and efforts are directed towards developing standard CAMAD tools for computer networks to be widely used as it has already been done in circuits. Probably, you have heard of SPICE which is very well known today for the purposes it has been used extensively. A knowledge-based system will play an important role in developing CAMAD tools for communications. (I hope I am at the right time.)

SHANMUGAN: I want to put up one of the viewgraphs I had in the introductory portion of my talk this afternoon. By way of summary, I am sure all of you have come to the same conclusion that we do have a variety of CAD tools available at all of these levels, starting at the lowest level circuits to links and networks. The CAD tools of CAD environments for transmission systems engineering is reaching a degree of maturity now and I hope that within the next year or two we will have standard tools or environments like BOSS available at this level. At the network level we are still at least five years delayed in terms of developing any kind of a standard framework for doing computerized modeling analysis and design networks. Incidently, someone said that CAMAD stands for computer-aided madness. One of the important problems that came through all through today was the lack of integration between CAD tools at these various layers. But there are some common trends that emerged during today's discussion. I want to point out a few things that I think are common trends at all of these levels, i.e., the use of hierarchical tools, more use of graphical front-end, also emphasis on transient as well as steady-state responses and so forth. So as you think over what went on today, I am sure you will be able to conceptualize some of the comments that we talked about today.

I have one announcement to make. Carl Ryan from Motorola has been kind enough to bring his CAD tool for communication link simulation, he will be very glad to show it to you today or tomorrow. Carl is sitting in the back, the machine is set up in the corner over there. With that, with Bob's permission, I would like to adjourn for the day, unless there are some short

questions.

HUTH: I have been playing with simulations for about 25 years now and what sort of bothers me was this idea that we are going to put novices in front of simulators or whatever tools or environments, and then let them go running away, while we put all the gurus off in workshops and whatever they do there. I guess my biggest concern is that I believe anyone can sit down and do a simulation with enough help to get going, and these kinds of tools help. What bothers me is what do you do with the results once you get them because my experience with novice enginners is they will come and show you these great results, which when you interpret them, may or may not be valid, and it is this validity of results that I am concerned about. And I also have a problem with "expert" systems because I was given a demonstration last week of an expert system, and it was not along this line, but the point was that there was an expert that had been used as the knowledge-base. We had three or four experts standing around watching this happen. They all disagreed with the expert on the computer. Not only that, but they disagreed with each other. And the problem is, most of these solutions in these kinds of results are resolved by confrontation, they are resolved by people really thinking about and getting deep into their thoughts, what does it all mean. I don't see that happening in the computer easily. I guess I am concerned somewhat with this idea that novices are going to do all this, and we are off doing whatever.

SHANMUGAN: I think you will still need a true expert to try to interpret all these opinions and interpretations.

HUTH: I think it is not a confrontation problem, there isn't even one expert in any problem, or even one expert system. There is a problem of trying to understand, especially when you work with new systems. If you are working with a system that has been around for a long time, we are talking about one-zies, two-zies, where they are new systems, we really don't know the answer until we get done, it's not clear cut. And these kinds of tools are helpful, but they are not the answer, and I don't see novices doing this.

MOUFTAH: The word expert, as I said, should not be used as "God on earth," or "God on the computer." There are many experts. If you put these experts together, we are not talking about the computer or the expert system. If you take these expert people in an area, on a table, and let them talk impulsively, they will conflict with each other possibly. Now the same thing happens if they will talk about an expert system computer as what you have seen, there was a conflict on the so-called package expert system. It doesn't mean that the so-called expert system is not a reality.

HUTH: My point is not that expert systems are not valuable, my concern is the idea that this expert is going to help everything and all of a sudden a novice can do everything, and in fact, isn't true when you come to real life problems.

BALABAN: Maybe an expert system is good for only routine things and not for new systems, where even the expert doesn't know what to do. It is an expert only for advice for new systems or for unusual systems, and not for routine things.

HUTH: Well, you see, that's what engineers do, they do new systems. They don't even bother to talk to engineers about old systems.

KUROSE: I've already said I'm sort of on your side of the argument. Half the

people in my department are in artificial intelligence, so I'll presume to answer your questions from their viewpoints, since I've asked them the same questions. The first thing they would tell you is your expert system is obviously only as good as the expert who caused it to have this knowledge in terms of associative rules. So I would point out to you that if things are being done by confrontation, well then, you're probably not perfect, and you might make mistakes just like an expert system makes mistakes. So you might mislead a junior engineer just like the expert system.

REY: The second thing someone in AI would tell you is that when you come up with a result, one of the nice things about AI system, the one that implies forward chaining, is that they can explain to you what the sequence of steps they went through in terms of the reasoning process, to give you a specific result. I think that can be a very valuable process because an engineer can say, "Okay, this falls in this, this falls in this .... Here exactly is the point where I disagreed, and maybe you can go in and ...."

BALABAN: That is part of an expert system. You can vector the system to make different decisions.

REY: I think we are being narrow on how we use these tools. There are management systems out there, there are control boards, so when a novice engineer goes and does design, he uses this tool, and he is very productive in doing it, but his results then are approved by his next level manager and looked at by his counterparts, and taken to review boards, and the thing is, he's more productive using these.

USC-CSI WORKSHOP ON ADVANCED COMMUNICATION SYSTEM ENGINEERING

Proceedings of Session Three: The Status of Optical Signal Processing

DUPREE: Good morning. This session is on "Status of Optical Signal Processing." My name is Jim DuPree; I'm with TRW in Redondo Beach, California. Our panel this morning consists of Bob Gagliardi of USC, Bill Steier of USC, and Dan Sullivan of TRW. We're going to branch off into something new. Yesterday we agonized over the complexity problem in dealing with millions of elements; at such a point along the path something usually changes -- a breakthrough occurs or people find easier and simpler ways to do things. That's our objective today.

I'd like to take a moment to thank the organizer, Bob Scholtz, and his colleagues and staff at USC for giving us a really beautiful spot to relax and brainstorm among this beautiful scenery. We've had a great time -- we've had magic in the evenings and "fortune telling" during the day. Our modern fortune-tellers use expert systems like BOSS, which is better than tarot cards at predicting how things will turn out before we start bending tin. But even with such tools, there will always be a place for the comm-system engineer to make applications of the new technology, to broaden our thinking, to learn new physics, and to find new tools to integrate into the system which keeps the comm-system business growing. And that brings us to the overview and motivation for this session.

In the satellite communication business, we're facing three constraints: (1) There is limited frequency spectrum and we have to learn to use it a little better. (2) The geostationary arc is getting congested, so we can't put up unlimited numbers of additional satellites. (3) Users want their own channels, they want to be connected, they want to provide on-site terminals and run their own comm business -- but they don't want to pay exorbitant costs. So we have to find a way to deal with this. We are finding that we have to put up more complex, higher frequency payloads, and in doing this we have to use them efficiently so that we will be able to optimize the connectivity, provide people with the rates that they need, allow them to be connected to any place, and provide them with reasonable throughputs.

SSTDMA is a coming trend in satellites. The requirements are microsecond switching times, 100 microsecond dwell times, millisecond frame rates, and bursts of 100, 200, and 500 megabits/second. In order to provide the users with flexible rates, we combine FDMA with TDMA on the uplink. By allocating FDMA channels in various chunks, the user can buy as much terminal as he can afford or needs. In the higher frequency images, we have to deal with rain, so adaptive coding on the up- and downlinks would probably be used in the event of rain. The payload has to adapt, either code and decode or not, depending on whether a user requires it. This adds up to a lot of complexity, but a lot of flexibility. The question is, "How can we provide all the channels on the uplink and a TDMA downlink?" You have to be able to separate the user's channel, but microwave filters would be quite heavy and the processor would require kilowatts of power, which is very expensive in orbit.

One approach to this is to go into more VLSI. However, the problem with VHSIC and wafer scale integration is that we're approaching limits on CMOS technology, clock rates, and chip size which ultimately mean that we can't continue to make things bigger and faster. Everyone seems to have one of these doomsday slides, so I'll show mine. [Laughter]. My fortune telling, my prognosis, is that feature sizes will continue to come down but ultimately will begin to run into production problems and the planned processes will probably saturate at 0.2μ in the year 2000. We can expect some commercially available wafer scale chips in the early 1990's. The numbers of elements per chip can continue to climb, but of course there are going to be limits on that because by scaling the chips you don't really scale the total processor power -- you just save on the drivers between chips. Pin connections are going to be a problem with that too, and so we will probably find ourselves making connections with holograms. This will become difficult to control.

We propose an alternate path: to begin looking at optical signal processing as an alternative. We may not clearly see the path to get there at first, so we must explore the potentials of optical signal processing. A logical way to introduce optical signal processing into the transponder is outlined here. I'll try to identify and develop optical signal processing functions which exploit the inherent advantages of optical signal processing. We take a time-varying electrical signal in a wire and, by means of a spatial light modulator transducer, convert it to spatial variations of a light wave and operate on it with spatial filters. By converting to a space variable, one can still do Fourier transforms and integrals and convolutions and correlations, but the operations are done in spatial coordinates. Optical computing is literally done at the speed of light. You can do a multiplication in the time it takes light to travel through a mask. Parallelism -- 2-D spatial light modulators could easily handle a million points. We might have spatial light modulators soon with kilohertz or megahertz processing rates. Even with a kilohertz rate, we could do a gigop per second.

However, the millennium is not here yet. We've been predicting that optical signal processing would change things ever since Lou Cutrona's work back in the 1960's, when we just needed some real-time film. Today, the problem is that we just need some real-time film. The early experiments to illustrate the principles used photographic film, because SLM's were not available. Today, we have some good SLM's being developed in the laboratory, but they are nowhere near the sensitivity and resolution of photographic film. But, of course, the films that are available today have progressed enormously since Matthew Brady lugged his darkroom in the back of a wagon, making colloidal plates in the field. Modern fine grain films have extremely high resolution, extremely low grain, and fast speed. But today's commercially available spatial light modulators have the sensitivity of fast enlarging paper, which would require long exposures in very bright light. And the resolution is only 20 or 30 lines per millimeter, compared with several hundred lines per millimeter for film. Thermoplastics do have extremely high resolution and are a very promising, though low sensitivity, technology.

Optical digital computing has been coming along fairly rapidly, and there was a commercial venture to build fast digital optical computers using analog convolution to perform digital multiplication. The prob-

IC COMPLEXITY TRENDS

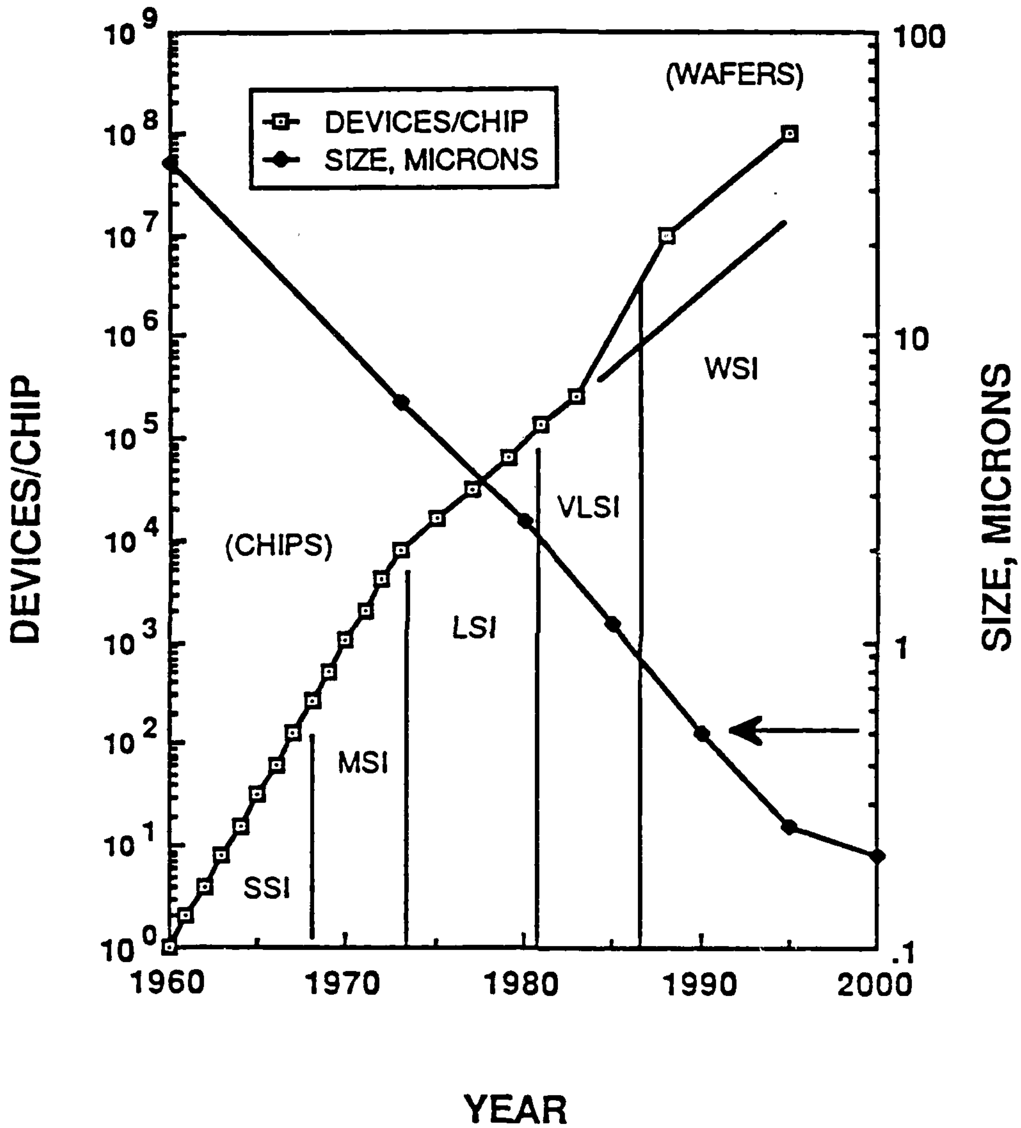


Figure 1. Historical Trends in Integrated Circuit Complexity and Feature Size.

lem was that the electronic support required to handle the throughput of the optical processor was so great that by the time they solved the electronic support problems, they needed only a small additional effort to make a very fast electronic convolver and do away with optics altogether. So those are the realities that we have to keep in mind in the present stage of optics development for signal processing.

We will try to present things now in a logical flow. Bob Gagliardi will talk about onboard switching and SSTDMA, and the prospects and potential of optical switching. We believe that, like any SSTDMA applications, a good optical IF switch might go a long way in implementing the onboard processor. Bill Steier will give an overview of optical information processing, and I hope he'll address the technology needed there and the status of the devices available.

Dan Sullivan will talk about analog optical processing of RF signals, and I think that you're going to have a real treat when you see some of the ideas that he has lined up. During his talk, I suggest you look for some of the key operations. How do you handle phase and frequency modulation and demodulation? Those are nonlinear operations. The good news about photons is that they don't interact with one another, so complex interconnection schemes are possible without interference. The bad news about photons is that they don't interact with one another, unless you happen to put them into some exotic materials. Photorefractive materials are capable of allowing a light beam to interact with another light beam. Then you should pay attention to the way he handles the conversion to and from the optical waveform as well as the architectures. I think this will be new material that you haven't seen in the overviews of optical computing or other applications of optical signal processing.

I'm going to turn this over to Bob Gagliardi now and we'll get started on the panel. We have only three panel speakers, so perhaps 20 minutes or half an hour each would allow us plenty of time to have a break and a question and answer session after we're done. I encourage you to make notes and ask questions because we think that this is a very exciting new area and there's something that we should all be paying some attention to.

GAGLIARDI: I think that we're all aware of the high speed data capability of optics, both fiber and laser. The point that I wish to talk about is slightly different. Specifically it is the possibility of using optics to do signal processing onboard a satellite. My hope is that we'll stir up some interest amongst the audience and perhaps even indicate some design directions for our research and for the optics world as well. One of the problems is that while there are advances being made in optical processing on the one hand, a good portion of the RF satellite people aren't aware of what's going on. Likewise, the RF satellite people are pushing for more onboard processing, pushing to the limit of what electronics will do, without even noticing what can possibly be done or can be extended by using optics itself. I want to focus on the satellite switching concept, because I think that's where optics could make the most immediate impact and possibly improve the systems.

In the satellite switching concept, because of the interest in using more spot beaming onboard satellites, the trend is to try to get more separate antennas, more multibeaming, and spot beaming onboard satellites. This puts us into a satellite

switch type of mode of operation. If you have a spot beam, you know that you can get more power in the beam, more concentrated EIRP, and therefore higher power levels. It also helps the LPI problem, provides a frequency reuse beam right next to another, for which you can use the same frequency bands. So there's a constant tendency to push towards multiple spot beaming onboard satellite. This immediately leads into a signal processing type problem, which I've kind of sketched out in this first graphic. [SLIDE #1] Roughly you have five beams on the left side, corresponding to uplink transmissions; we have downlinks, five beams, on the right hand side, and we need some kind of onboard processor to do the interconnection, basically to tie together and distribute the uplinks amongst the downlinks. The onboard processing has primarily the requirements which are shown. We must have isolations, since these are RF waveforms that are being moved around onboard the satellite. It is important to make sure that they are completely isolated from each other without interference. There's a connectivity problem. Generally we want to make sure that every downlink beam can listen to every uplink beam, so there must be some type of interconnectivity between the two. And of course there's always the weight and power problem. You want to make sure that you're not building up the weight and power requirements onboard the satellite as well. These particular requirements lend themselves nicely towards optical type of solutions if these are possible.

Let me just back up a step to review some simple types of satellite switching that's currently being done. This graphic [SLIDE #2] shows a satellite switch FDMA concept which is used. On the left hand side three uplink spot beams are coming up with three different frequency bands divided up amongst the same frequency bands of each one. They're all using the same total frequency band but they have separated out the frequencies that go each way. Then by filtering them off, using the channel filters and going through a switching matrix, one can then distribute the uplink channels to each of the corresponding downlink spot beams, shown here. The same thing is being done on the other spot beams. By properly keeping track of where each band is, for example, all the darkened frequency bands can be sent over to one particular downlink, etc. for the others. We can now see that the requirement in the center, is for a microwave crossbar switch which will basically allow the bands to be separated out, to be basically guided over to the corresponding downlink with little amount of interference. These are RF waveforms now which are passing right by each other, and you would therefore like to have the isolation in there so that you will not get interference. Also you have to worry about interference from perhaps other electronics onboard the satellite itself in the C-band or K-band subsystems. You may have carrier frequencies generating spurs, you may also have interference from another satellite, or maybe even nuclear radiation type of effects as well. So the isolation problem is quite important in each of these devices. The connectivity is provided by means of the corresponding switches. Some of these microwave switches tend to get heavy if you start cascading many of these together. This is basically what is done in the satellite switch FDMA format today.

This graphic [SLIDE #3] shows a TDMA type of format, where we're doing satellite switching TDMA. Remember that the uplink now reaches spot beams on the

left hand side corresponding to a TDMA concept. These are slotted type of transmissions, all using the same band, all coming up at the same time. What's done is to connect with a switchable gate matrix. Spot beam No. 1 could connect to downlink beam 4, etc.; 2 to 5, 3 to 6 during one window time. This then allows the corresponding stations in one spot beam to listen to stations to which it is connected to on the upbeam. Then during the next window time when the switch is made, the system is reconfigured, and reconnected back up again, so that as shown here 2 talks to 5 and 3 talks to 6, etc. Conversely you then go to the next window where 1 goes to 6, 2 goes to 4, 3 goes to 5, etc. This graphic shows a matrix of all the switches. So here you need a microwave gate in which you reconfigure each window time. Again, these are RF waveform which are passing through this gate matrix here. The faster you can do your switching, means the faster that you can revisit each of the corresponding downlinks. A given downlink station therefore sees the uplink beam more often, depending on how fast you can do your windowing operations. Initially Satellite Switched TDMA was done at a millisecond rate, but in the latest generation INTELSAT switching is being done at microsecond rates. If we can push towards nanosecond type of switching, if we can switch at nanosecond rates while we're uplinking with 500 megabits per second, we could literally switch out bits one at a time with each of the corresponding downlink states. That's real time distribution of the uplink bits that saves all the buffering problems that generally go along with a TDMA type of operation. These systems do exist, they're not just being proposed.

This graphic [SLIDE #4] shows a West Star system, going back in 1984, and shows spot beam on the left hand side uplink and downlink, and a 4×4 switch that was controlled, as I said, at a millisecond rate. INTELSAT 6 I think is up to six or eight switches going at microsecond rates. It's a concept which exists at the electronic level. The question is, "Can optics help us in this type of processing?" There have been some advances made that I think are interesting. About a year ago I think, Bell Labs announced that they took an RF carrier, a C-band RF satellite carrier, and put it on the laser diode. They intensity modulated a laser diode with the complete RF spectrum corresponding to a C-band transmission, put it on to a fiber transmission, photodetected it, bandpass filtered it, and recovered the RF carrier. They took an RF carrier from one point to another using a relatively simple system. A laser diode, a fiber and a photodetector. They were able to move an RF waveform from one point to another. That's a really important accomplishment. You can take an RF satellite band, and move it from one point to another with a simple fiber link of this particular type. We're always talking about the impact fibers might have on groundstations. Now you literally can have, instead of a whole series of independent groundstations, one receiving station and then simply use the fibers to disperse that whole RF spectrum. This may be helpful from a satellite switch concept as well.

Let's go back to our FDMA format and consider how we might for example use a fiber type of assembly to do the satellite switching FDMA format that we mentioned before. The left hand side of this graphic [SLIDE #5] shows the spots coming up ... bandpass filtered to each of the individual bands. And now we simply put each of the filtered bands on a fiber, and it's probably easier, since the bands are

smaller than the complete 500 megahertz satellite band. In fact they may be as low as 36 megahertz of information. We modulate it directly on a diode and cross the fiber to a photodetector, demodulate it back out again at the other end of the satellite, and put it on the corresponding downlink and send it down. Then of course you have the crossbar coming in with the other channels from their spot beam; same thing, connected back up again so you're getting the same connectivity that we had before. I showed separate photodetective elements but sometime in the future we might actually be able to sum the optics and then do a single photodetection and cut down on some of the actual optics that are involved. So we get the same type of operation we had in satellite switched FDMA. However, we're doing it all with optics and now we have confined the waveform being transmitted from the uplink beam to the downlink inside the fiber. It's completely oblivious to other electromagnetic radiation. There is almost zero cross-coupling between the two. There is no effective outside radiation at all as far as the individual fibers are concerned. It's almost like running the RF directly from one point to the other with almost no interference at all. The connectivity is maintained and we've done it with relatively simple components, as far as weight and power are concerned. It appears that this type of a switching connectivity would give us advantages at least toward solving problem areas mentioned before, the isolation problem in particular. We can do the same thing with SS-TDMA. We can make use of that same fiber concept in terms of a satellite switch TDMA concept as shown on the next graphic [SLIDE #6]. You'd have to have a way of realigning the fibers in terms of the connectivity that we mentioned before. So again I show spot beams on the left, spot beams on the right, and these uplink spot beams are modulated directly onto a diode, cross through the fibers to a photodetector at the output. Now if I can switch my fiber connection, that is if I can have a form of an optical switch that will allow me to transmit my optical beam, instead of on a fixed crossbar but not to these other down links, and vice versa. A simple 2×2 type of switching operation that can be done would therefore allow me to interconnect these two. I've shown here a 2×2 system here, so obviously you've got to build these 2×2 switches up into trees of a matrix operation. I can now switch these at optical rates, (Bill Steier later on will talk about some ways of actually doing optical switching of optical beams), taking two beams and switching them from one point to another at optical rates, that is, at nanosecond rates. We can get close to what we mentioned before, an actual bit by bit satellite switching done onboard the satellite itself. And again you can think of some type of a clocking mechanism that uses fiber switching for us so that we can switch the photodetectors as we've shown here. Again, everything is inside a fiber, there is almost complete isolation of one fiber to the other, and of course a relatively simple system again as far as weight and power is concerned. There's another way to do the same thing. We don't necessarily have to switch optical beams. We said that in the fiber system you want to be able to switch from one fiber to another. You're actually moving an optical beam by means of your interconnects. It may be possible instead to do something like this [SLIDE #7], for example. Rather than do the separation of the fibers itself, you might think of one single fiber for all the uplinks, and putting each corresponding uplink on a diode of different wavelengths, going to the

common fiber with the different wavelengths, and doing some kind of wavelength splitting, some kind of a diffraction grading type of operation, something that will allow me to bend the beam proportional to the corresponding wavelengths that we mentioned before. A dichroic mirror, diffraction grating, or maybe even some kind of a brag cell arrangement that might allow us to do the same thing. Instead of switching fiber beams, we simply switch wavelengths; if I can somehow change the wavelengths of my diodes I immediately change which of the input goes to which of the corresponding outputs. So I can actually switch on a wavelength basis, rather than doing switching of the optical beam itself. This of course can be done even faster and easier.

There is still another way of doing the same satellite switch TDMA concept that we mentioned before. We don't have to deal with straight analog systems. We can do it with digital systems, and there have been some interesting work in fiber digital systems that I think also applies in this particular case here. This [SLIDE #8] shows a simple on/off key fiber system. You've probably seen these presented before. You have data bits being generated, you trigger a laser diode and the laser diode generates a pulse, goes over a fiber, is photodetected, filtered, and produces the pulse waveform that you sample and decode. Therefore you're transmitting the bit from one end to the other. It is a standard on/off key type of system, and if we're going at 10 megabits per second that means the diode must pulse at that rate. Present diodes have pulse widths on the order of 10 nanoseconds or so, so the diode isn't working that hard. However we're not taking advantage of important properties, mainly the high bandwidth the fiber has, and the relatively high bandwidth the laser diode has in producing the narrow pulses used for transmission. The extension to this is into systems shown here on the next graphic [SLIDE #9], which is a pulse sequence on/off keyed doing exactly the same thing, only operating with a pulse sequence. The bits come in and again modulate the laser. Instead of producing one pulse at the output you produce a sequence of pulses corresponding to a fixed pattern. During the corresponding bit time you then send the pattern to the receiver photodetector, as I've shown here, and electronically correlate up those pattern pulses to form a correlation pulse similar to that transmit initially, and then make your big decision: is there a bit there or not due to on/off key concept. We are simply using a sequence of pulses instead of one big pulse, and you can do this because the fiber allows you to transmit those much narrower pulses at the same pattern rate that you had before. The diode isn't working any faster; it's still putting out a single pulse that is encoded into this multiple set of pulses. But the photodetector has to also have high bandwidth in order to operate at the pulse pattern rate.

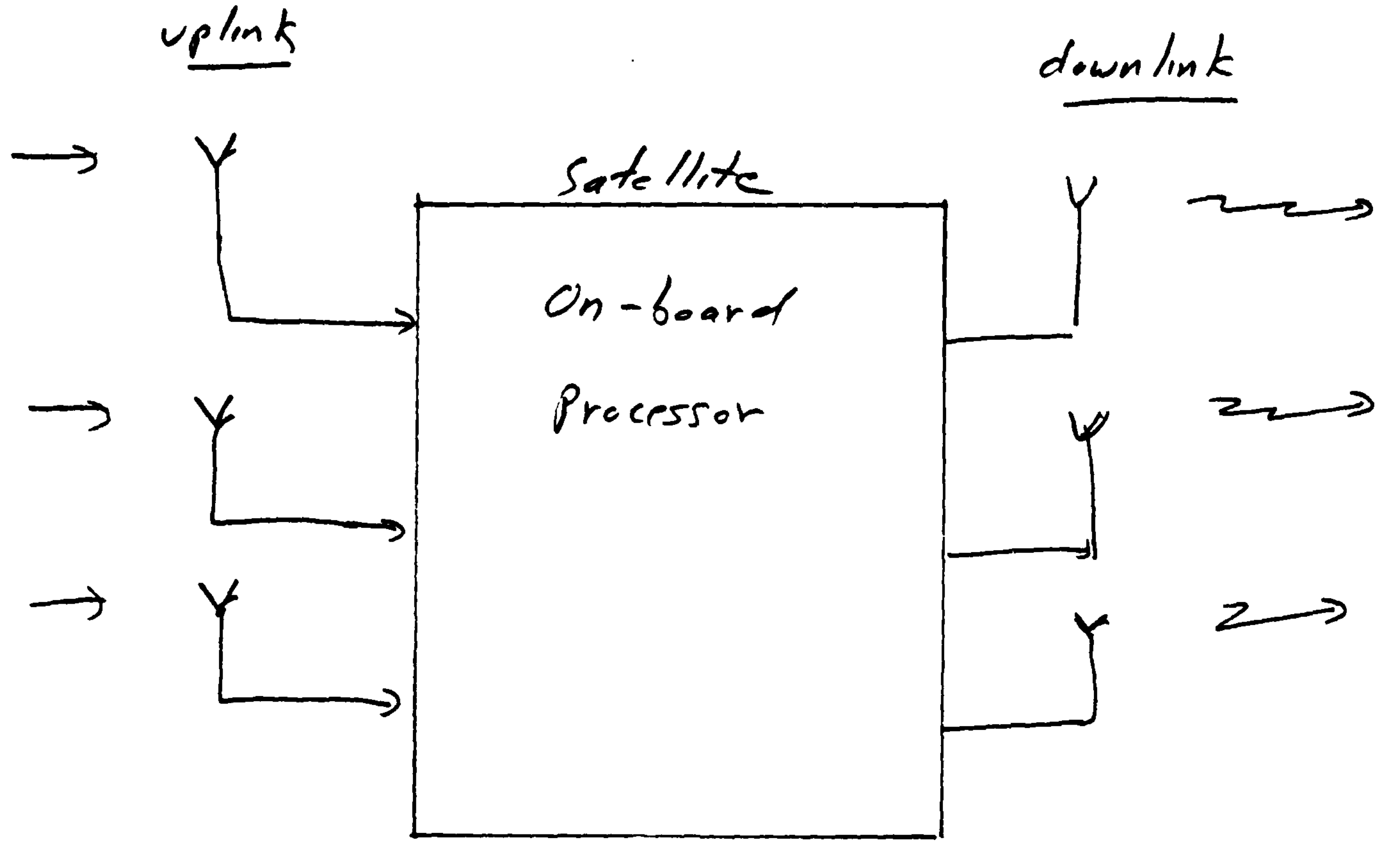
Another way of doing the same thing, which is of considerable interest now in the optics area, and that I think is a really important result, that the optics people don't seem to be excited about and communications people aren't aware of, is that you can do optical correlation. You can do all the correlation optically. You can take the bits and encode them into a sequence of pulses again, transmit them over the fiber and then optically correlate, optically integrate up the pulses into one big optical light pulse, and then photodetect that single light pulse. Then of course do your sampling at the output to get back the on/off bit that we mentioned before. Therefore you

are allowing your photodetector to operate at the bit rate rather than at the pattern pulse rate that you actually send. And you're taking advantage of the large bandwidth of the fiber itself.

[SLIDE #10] Optical correlators can be constructed several ways. One of the popular ways in the literature is by means of simply delay lines. Your pulse sequences are pulses spread out in time. Any system that can put back those pulses back together is equivalent to an optical correlator. So a tapped delay line can be constructed by just combining the various pulse delays. This is done by means of a fiber bundle in which you feed the light waves in, split it off at the output, and you have delays of different amounts inside the individual delay lines that sum at the output. By properly designing those delays you'll sum all those pulses on top of each other and get one combined pulse at the output. Again, a relatively simple system, and optical correlators have been built. You're primarily concerned with the length of the fibers that you need to get the required delay. In fact, you can actually use the correlator to generate the pulse sequence back at the transmitter, because I can use the same technique. We hit the input with the pulse, and then have the various delays to get the pulse sequence at the output. So I can do this then again at the transmitter and make use of an optical correlator of this particular type. These have been demonstrated, at least in laboratory, as relatively easy to design. The applications of that which spurs the interest in that type of processing is CDMA. You have on/off digital links as shown in this graphic [SLIDE #11] coming into multiple lasers, individual bit streams, and they each have their own particular code pattern as we mentioned before. They superimpose their code patterns asynchronously into a fiber, the fiber transmits it, and then you have an optical correlator photodetective for each individual pattern. Each correlator looks for its own pattern, and hopefully will not see any interference from the other patterns and therefore will build up to a correlation value that peaks. if the pulse pattern is present. Therefore with this way you'll detect a bit from a particular source. This is a typical type of spread spectrum correlation being done optically.

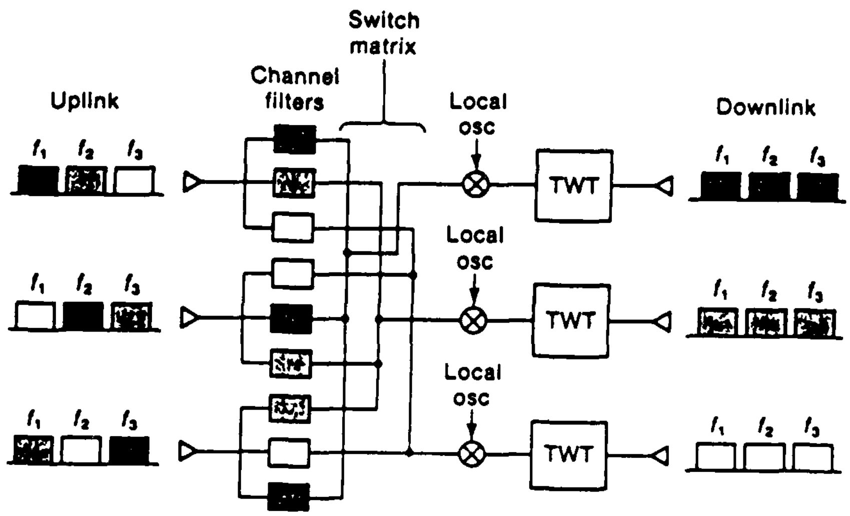
An interesting little problem is the code selection, where you place the pulses so that you don't get interference with each other, has led to some interesting coding problems. That is, trying to find codes with pulses symbols of 1's to 0's, that provide good interference properties, low cross correlation distribution between them, while satisfying constraints. The longer you make the code pattern, the longer the separation between the two outer pulses, and the longer is the length of the fiber delay you have to allow. That means longer fiber length. So you might put a constraint on how long you want the code, and then you can play with correlation values. Here are some equations that relate the number of different users you can get with code patterns with minimal amount of cross correlation in a sequence pattern of this particular type. Again, our interest is in doing satellite processing.

One thing we can immediately do, using the basic digital fiber concept that is being pursued in the fiber world, is to convert that whole concept to a PAM concept. We can take an analog source and sample it so that we actually produce samples of a waveform. [SLIDE #12] I show here some of them on the left hand side, we then encode into our diode sequence generator that we just mentioned, and actually code the amplitude of the pulses in the pattern.

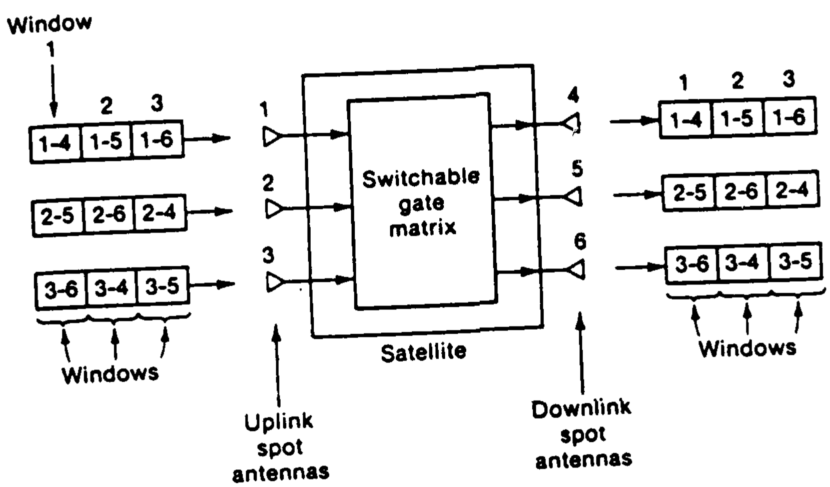


- Isolation
- connectivity
- wt/power

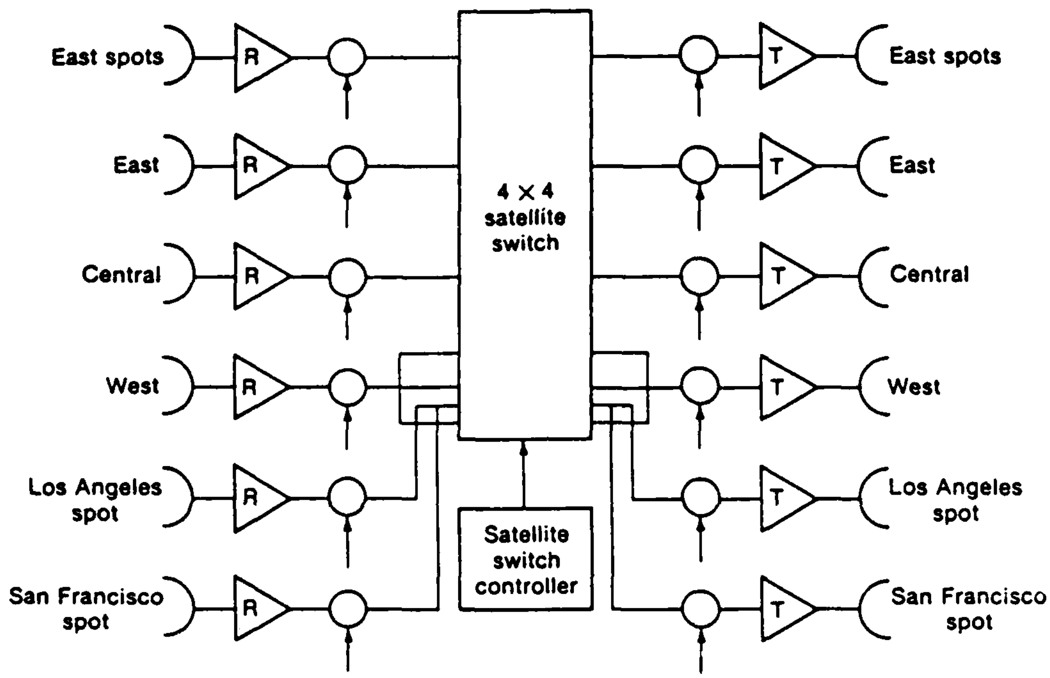
SLIDE #1


Switch matrix
Channel filters
Uplink
Local osc
Downlink
f_1 f_2 f_3
TWT
TWT
TWT

SLIDE #2

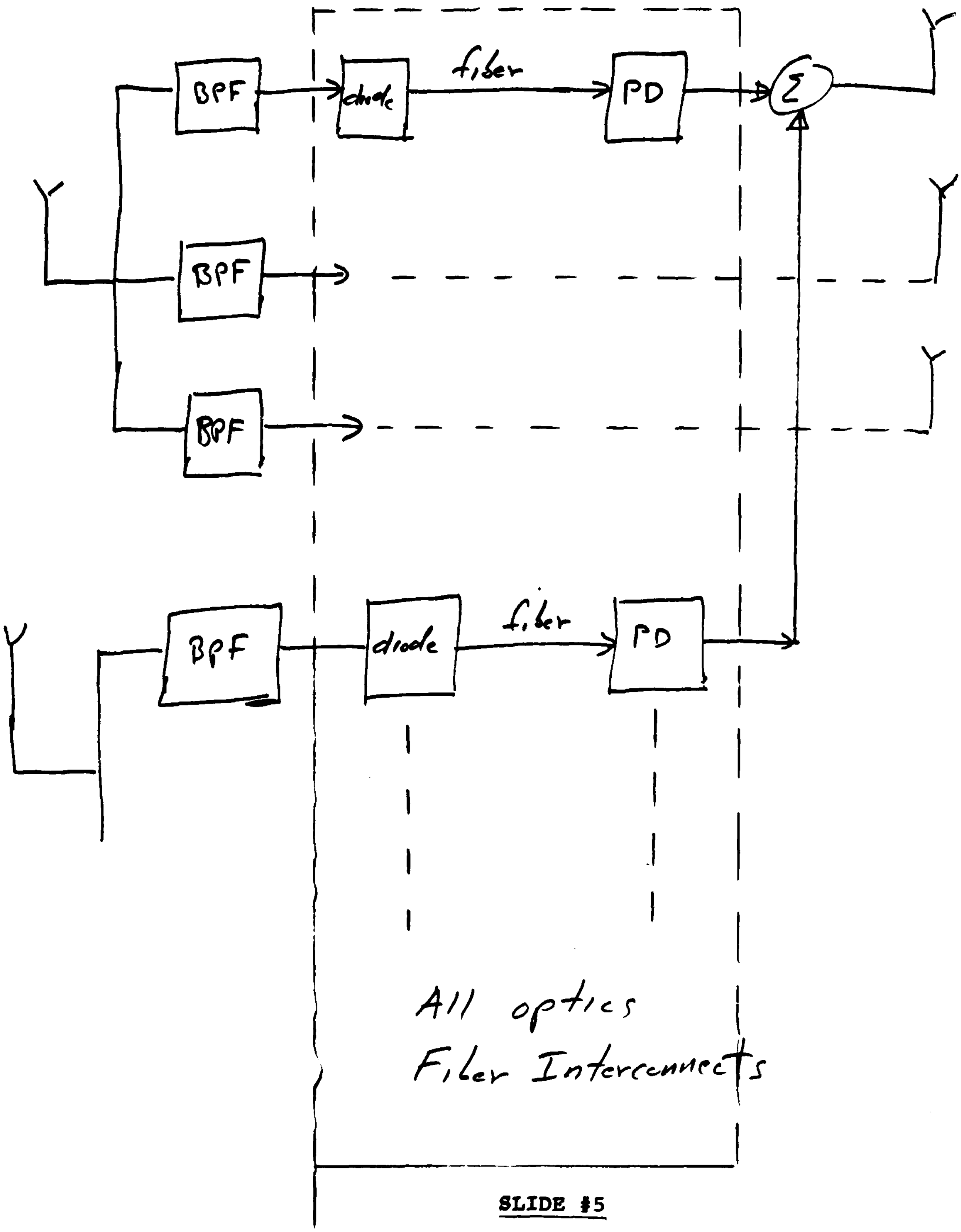

Window
1
2
3
1-4
1-5
1-6
2-5
2-6
2-4
3-6
3-4
3-5
Windows
1
2
3
Switchable
gate
matrix
Satellite
4
5
6
1
2
3
1-4
1-5
1-6
2-5
2-6
2-4
3-6
3-4
3-5
Windows
Uplink
spot
antennas
Downlink
spot
antennas

SLIDE #3


East spots
East
Central
West
Los Angeles spot
San Francisco spot
R
R
R
R
R
R
4 × 4 satellite switch
Satellite switch controller
T
T
T
T
T
T
East spots
East
Central
West
Los Angeles spot
San Francisco spot

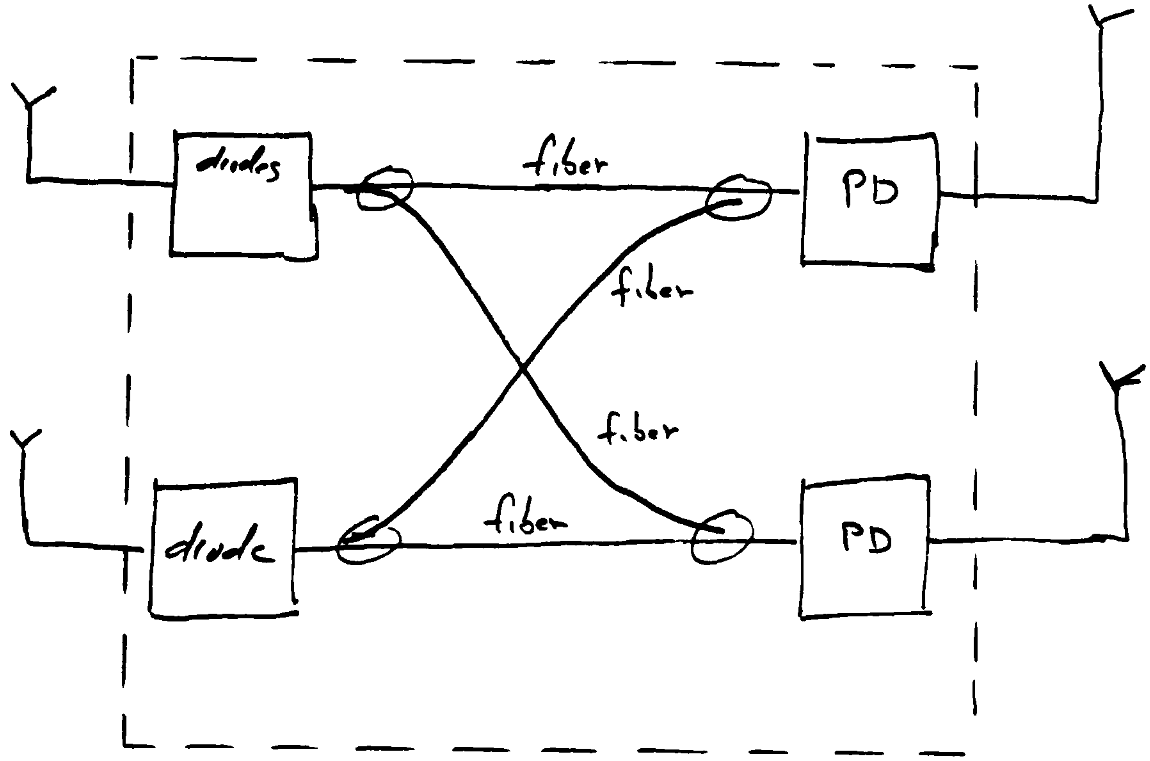
SLIDE #4

Fiber SS/FDMA



SLIDE #5

Fiber SS/TDMA

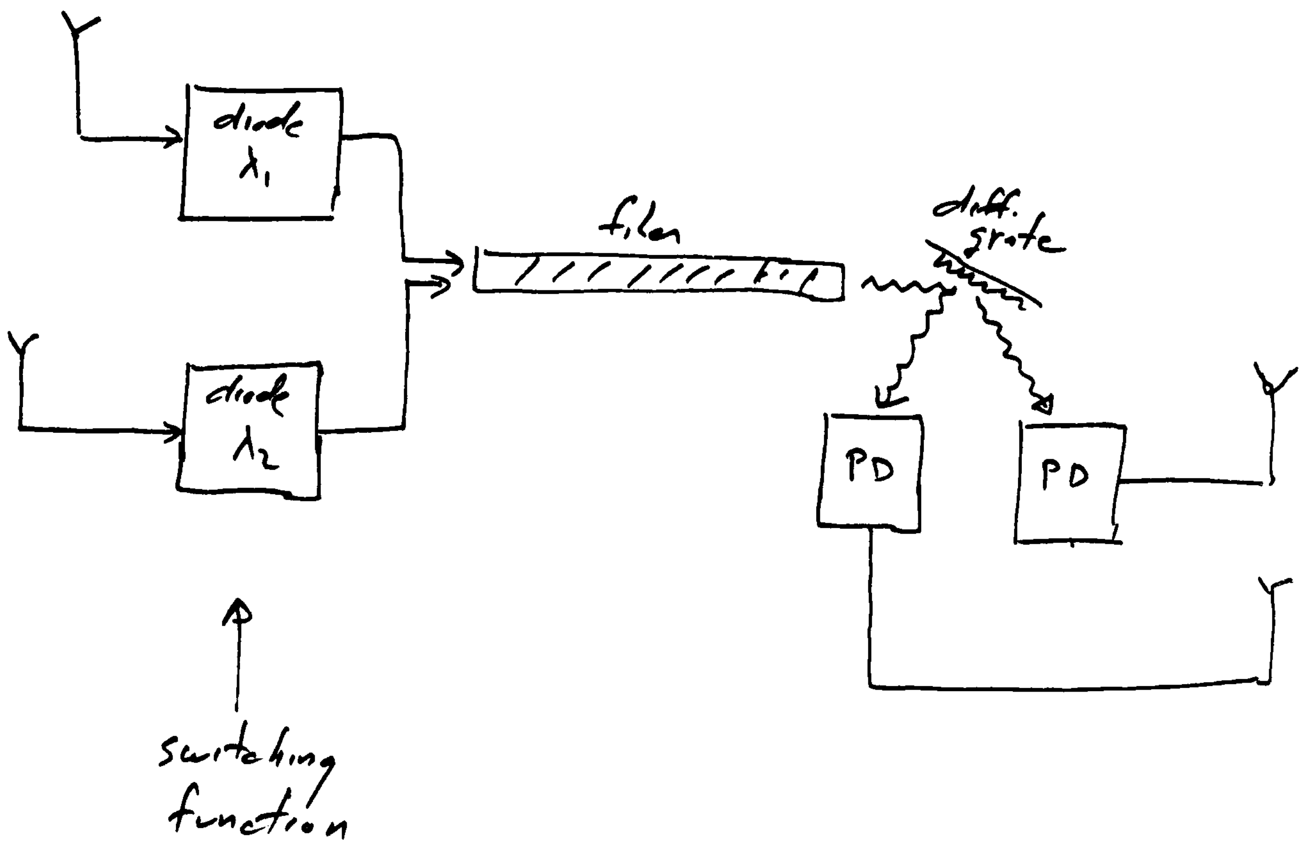


All optics
Switchable Fibers

SLIDE #6

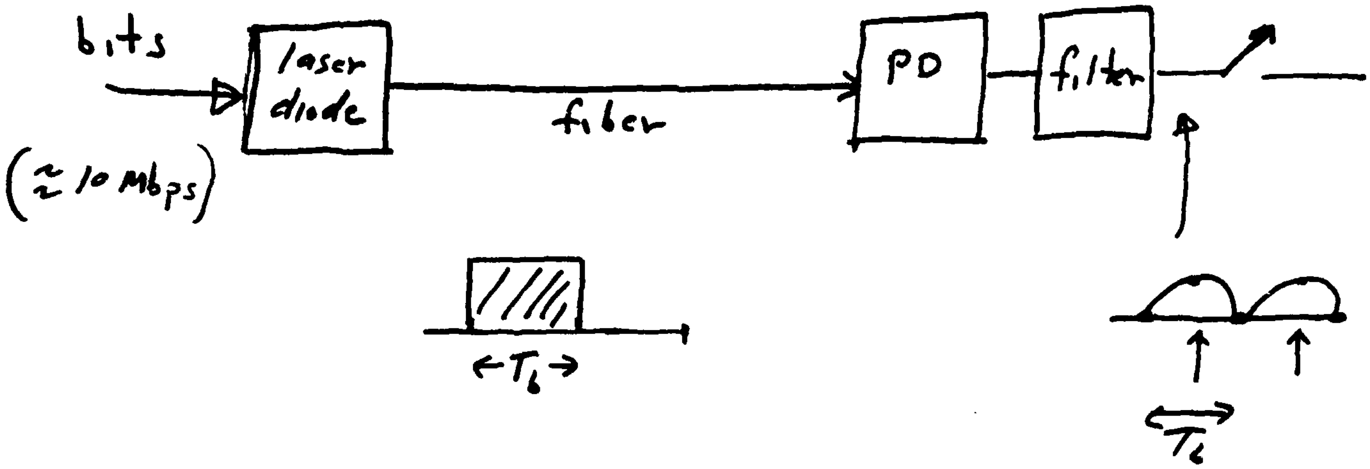
Optical SS/TDMA

Wavelength Switching



SLIDE #7

OOK Fiber System

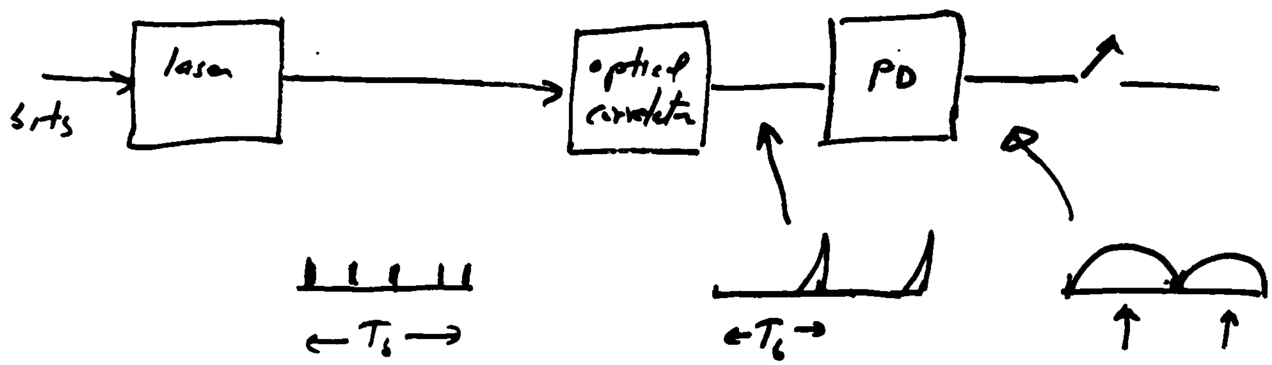


Key parameters:

- PE
- bit rate
- sync

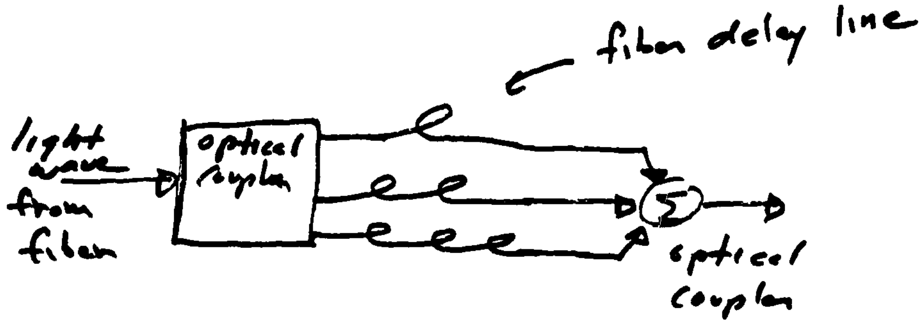
SLIDE #8

Pulse Sequence OOK



SLIDE #9

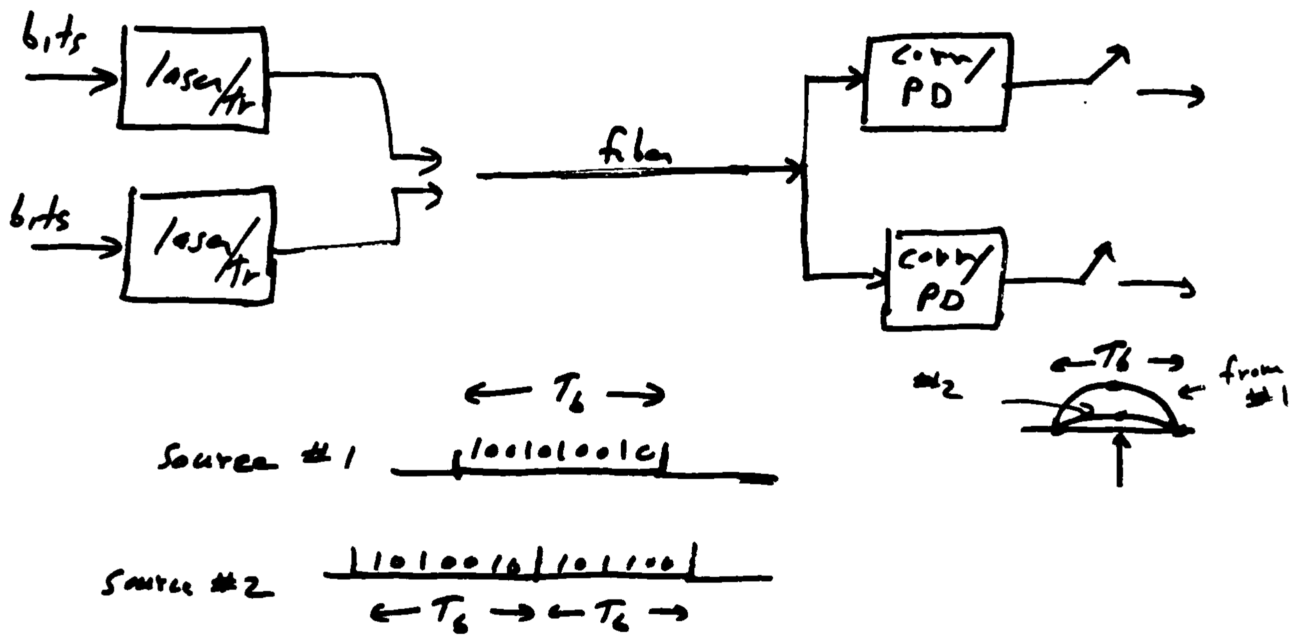
Optical Correlator



• can be used to generate pulse sequence at transmitter.

SLIDE #10

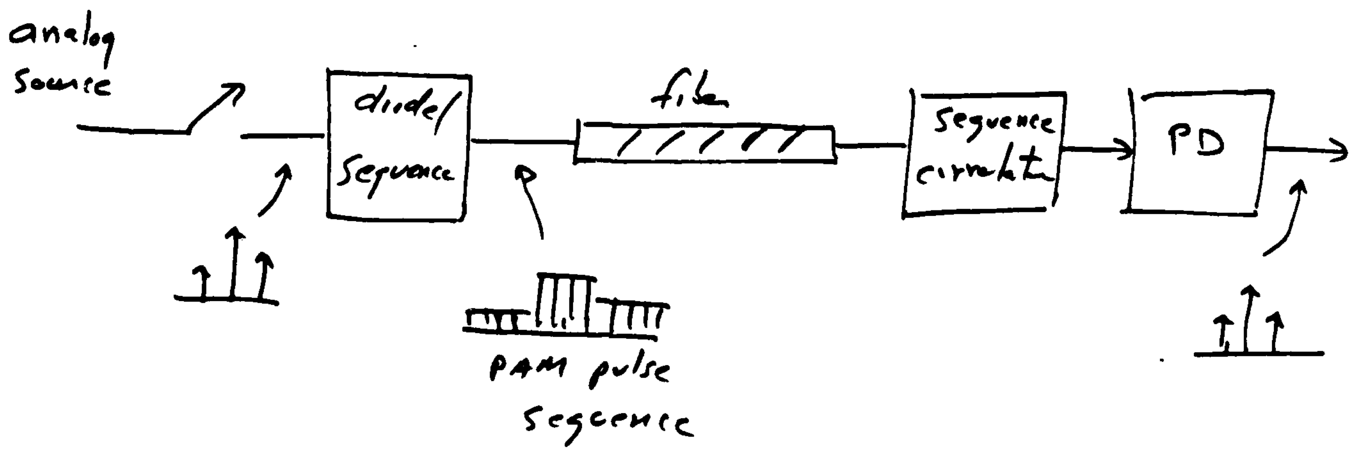
CDMA - OOK



- optical correlator detects only the proper code

SLIDE #11

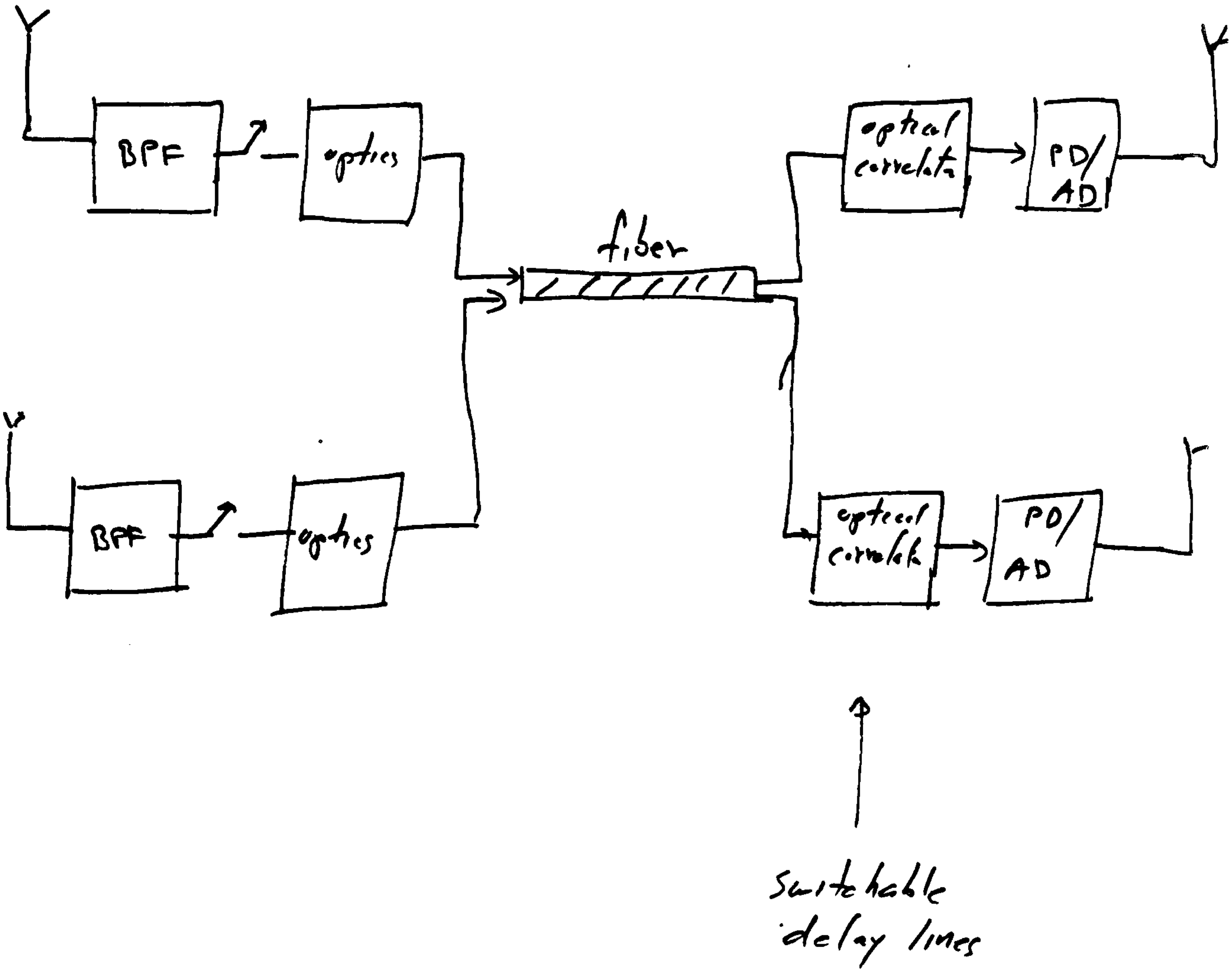
Coded PAM optics



SLIDE #12

Optical SS/TDMA

code switching



SLIDE #13

So you have a pulse sequence being produced that will have different amplitudes, based upon the sampling that we mentioned before. Transmit them over the fiber. The fiber has bands that can handle those pulses. Correlate now with the proper correlation looking for the proper sequence, and then photodetection to generate those same samples that you had before. I can therefore transmit analog information by the same digital encoding technique that we mentioned before. Instead of being an on/off keyed you're changing the amplitude of the pulses corresponding to the pulse sequence. This now allows us to basically apply that technique as we said before. This is then, Satellite Switched TDMA, making use of code switching as opposed to the wavelength switching that we mentioned before, or the fiber switching that we mentioned earlier. I can conceive of doing something like this as in this graphic. [SLIDE #13] Here are my spot beams on the left hand side, and my objective is still the same. I bandpass filter each spot beam uplink, sample to get an analog waveform or at least a representation of that, then go to the optics that we mentioned before; a laser source and a sequence generator to give you the corresponding sequence, and combine them all into a common fiber. I am on the satellite now, going from one end of the satellite to the other through the fiber system. I now have an optical correlator that's looking for the code of each of the individual sources. Each correlator then can, in a CDMA concept, pick out the code that it's looking for, and correspondingly get back its sample value, D/A convert to get back to the actual analog waveform. So we're doing again RF/RF transmissions, going through a simple fiber from one end of the satellite to the other. For the TDMA concept, we've got to be able to switch this with the connectors we mentioned before, at a corresponding clock rate. So, as we said before, we can try to switch the optical beam. Or we can try to switch the wavelength of the diode to take advantage of the wavelength separator. Here's another way. Now we can try to switch the delay lines. If we can actually reconfigure those delay lines so that a given channel looks for a different code, then I can basically accomplish the same operation as before. I can interconnect the left hand side to the right hand side, corresponding to a clock synchronization. This switching is a little bit different now. I'm not really moving optical beams, I'm simply altering delay lines. I'm simply adjusting the composite delay. Perhaps there may be some way of using nonlinear optics for this type of adjustment, or using the internal delay of a resonator of some type that basically readjusts this. If we can do this, again at the nanosecond rate that we're interested in, then we can conceive of actually doing bit by bit switching on a TDMA concept. So this is something that again might be worthy of exploring. There is progress being made in the pulse sequence design and these devices are fairly well established. This would be one way to extend it into the satellite switch TDMA concept. I don't think there is anything else that I wish to add and perhaps you might see ways in which we can do further operations to this. Yes, Ray ....

PICKHOLTZ: Has anyone built any of these optical correlators and ....

GAGLIARDI: In the laboratory, yes.

PICKHOLTZ: And what kind of parameters are they reporting, what kind of speeds? You're talking about what the delay lines are?

GAGLIARDI: I know the people at Columbia are working on it, and they have experiments being reported. I'm not sure what the delay, which they are reporting, is; I would say they're producing kilobit rates, using code sequences that may involve 7, 8, 9 pulses. Their pulse codelengths weren't that long, so the actual fiber delays weren't that excessive. Remember, it takes a foot of fiber to give you a of nanosecond delay, roughly, so that when you're building up long delays, on the order of microseconds or so, you will need long fiber bundles to produce this delay. There may be other ways of doing it. I indicate a fiber line; there may be better ways of getting optical delay. The numbers I recall are kilobits per second, a relatively low rate. The purity of the correlation, and the ability to correlate where the correct signals decorrelate out the incorrect signals, is pretty good from what I have seen.

AMOROSO: I have two questions. What sort of correlation properties among the codes would you be looking for? (What sort of correlation properties among the codes of a code division multiple access system) and also what is the performance of the optical correlator at the system's level, for example, bit error rates versus something related to signal and noise? Those two things ....

GAGLIARDI: Well, there's been analysis of that type of data appearing in the literature. That is, bit error probability with respect to the amount of correlation you have. It's fairly easy to see what happens. If you get more interfering correlations amongst your additional users, you build up correlation, and the error probability is based on the threshold which is selected between the two. There are some modifications that are being used, such as limiting the input so that you never build up more than a one pulse point interference per given slot time, for a given pulse time. So there are tricks of this type that can be inserted.

The amount of correlation you can withstand again is an interesting parameter here because you're using a 1-0 codeword, and selecting code symbols in a sequence such that you simply limit the amount of code overlap. With a little bit of thought you see that the key thing is the distance between the symbols here, and you want to make sure that you don't repeat distances between symbols to avoid the correlation buildup here. In the literature they talk about optical orthogonal codes, this seems to be the title that they're using. They're not truly orthogonal but instead have a correlation of one. You're bound to have a cross correlation of at least one as you slide one code by another. The problem is to keep it at only one. Those are called orthogonal codes. They're called optical codes because they use 1's and 0's instead of +1's and -1's. For those of you that are into PPN sequences, the difference here is that you can't use minus symbols, so you can't build up negative correlations. You must build up your correlation and your cross correlation with 1's and 0's. This result here that I previously quoted out of the literature shows that the number of members of the code set can be related to the codelength. That's how many code positions that you're going to allow, divided by the number of pulses that you're going to use in the code, and the latter appears squared in the equation. So you build up correlation higher as you use more code pulses, but the number of code set members goes down, and you can design with bounds of this particular type. People at USC are doing studies in this area, again in the mathematical point of view, trying to

find new classes of good codes. Of the codelengths with three or four pulses, they're getting complete solutions; that is, all the optimal solutions that produce orthogonal codes, with a cross correlation of one. So it's an interesting little problem that I think can be expanded here. It's entirely independent of the optics because it simply boils down to code setting distance.

AMOROSO: Well, I wanted to add something to the other question. How does the performance of the optical correlator in noise compare, let's say, with the performance of a pure pulse position modulation scheme in the same noise environment, let's say comparing average received signal power on the basis of equal average received signal power in the two systems?

GAGLIARDI: Well, it's hard to compare the PPM concept with this, because here your object is CDMA. You're trying to get many users into the same time frame.

AMOROSO: Well that could become pulse position multiple access .... Is the simple minded statement not true that I could just let the different users use a different pulse position or something like that, as opposed to a different pseudorandom code?

WELCH: Well, no. I think that he was contrasting ... he earlier mentioned a wide bandwidth photon detector. But then he went on to the narrowband, to the slow photon detector, and you can't detect with slow photon detectors.

GAGLIARDI: You're doing the correlation directly with the optics. Remember the correlation is done optically now, so the photodetector only sees pulses at the bit rate, not at the pulse rate.

AMOROSO: So are you saying that you're getting a narrow bandwidth rejection of noise with a slow photon detector? You actually reject noise as a virtue of its narrow bandwidth, or slowness?

GAGLIARDI: I don't think it's that obvious because you're summing up many pulse positions.

WELCH: You're integrating ....

GAGLIARDI: You're integrating the pulses up, and you're going to integrate the noise as well. So it's just like additive noise in any other type of interfering system.

AMOROSO: Okay, I understood that there are practical situations in which you can't think of it in the same way as you think of a microwave system, so it's encouraging to know, at least in this case, you can use microwave type thinking to reason it through.

GAGLIARDI: Yes, if you're making a Gaussian assumption on the output of the photodetector, then the standard CDMA Gaussian analysis applies. Question?

MOHANTY: How much will be the processing delay caused by switching to this optical correlation and separation detection? How much do you estimate that delay will be in processing time?

GAGLIARDI: Well these delays, you're only running the fiber 20, 30, 40 feet, so there's not long fiber length delays in here.

MOHANTY: How about optical correlation? Do you consider the delay lines multiplexors, aiding those?

GAGLIARDI: The upper limit is the pulse sequence time. You have to wait for all the pulses to sum back up again, so that's the upper limit of how long you have to wait; otherwise, I don't see any more delays, at least on the optics side of it. Of course when you get into the electronics

side, you may have to worry about some delay effects as well.

REIFFEN: A comment .... I think that it should be observed that when looking at the satellite payload of a system, one is still burdened by all of the uplink antenna hardware, uplink received and front end, downlink drivers and downlink transmitters, which generally overwhelm the signal processing weight in much of the systems which we're talking about. So we do have the potential of making significant improvements in the weight of the signal processor in selected systems, but one must appreciate that that's only a fraction of the total payload that one is concerned with here.

GAGLIARDI: Sure, yes.

SULLIVAN: The higher the frequency the lower those other other things will be. The higher the frequency, the smaller the antenna, the lower the power.

REIFFEN: Well, but not in terms of the weight of the, let's say, front ends or the travelling way to downlinks.

HUTH: I guess I have a slight counterexample of what you're talking about. There's a good example where this does apply, maybe to a NASA space-station because there you're trying to move the carriers, and various things coming off the air, substantial distances. Your other approach would be to mix them down, put them over co-ax and take them across, and so there is, when you can do the whole thing optically, are some examples.

REIFFEN: Certainly ....

HUTH: So that's a good example when you can do this kind of stuff. But when you're talking about ....

REIFFEN: That's strictly in fiber optic links ....

HUTH: Yes, but depending on what you're trying to do, you can use these kind of techniques also, since you're going to have to go on fiber anyway, you can solve some of the problems, solve more than one problem at a time.

SPILKER: In the microwave to optical converter in the photodiode, what is the drive level in the microwave energy required, and what is precisely the output of that optical diode?

GAGLIARDI: I'd have to go back and take a look at the numbers. I just don't remember them. This has been reported for direct modulation of the C-band directly onto the diodes. I don't remember any of the system's parameters.

SPILKER: I guess I was just thinking that those levels must be fairly heavy, which means that you have to have a fair amount of microwave amplification before you get to that point.

GAGLIARDI: Yes, most definitely with modulation, but you'd expect that you're not going to have a power problem because you're running over short lengths, at least for this application anyway. You're putting modulation on and taking it off almost immediately, so there's not much loss involved here.

SPILKER: No, I was really just amplifying on Barney's concern that you have to still amplify the signals at a fairly high signal level before they can actually get to the optical end of the thing.

WELCH: Do you know what the bandwidth was of that experiment? Is it everything from DC up to C-band, or is there a bandwidth involved?

GAGLIARDI: Well, the regular satellite bandwidth was used. So a C-band at 3, 4 gigahertz, 500 megahertz wide, is

put directly on the diode. I would expect it doesn't have bandpass filtering, but the upper end of the band is what you're concerned about, and that's in the GHz band.

MOHANTY: That Bell Lab experiment, is that just for C-band or for higher band EHF, or higher band? Is that the one that you said that delay ...?

GAGLIARDI: Well when you go up into gigahertz, you're starting to push the diode and the photodetector bandwidth. So when you start getting up in C-band frequencies, and K-band, you're asking for diode photodetectors operating at high gigahertz. In terms of gigahertz, that's kind of hard to do. With the C-band you can always downshift to 2 or 3 gigahertz, and then do your optics and then up convert that after photodetection.

MOHANTY: So you cannot do anything right now for the NET-EHF band?

GAGLIARDI: I haven't seen anything reported. You're beginning to get to the limit of what a diode can do modulationwise and what a photodetector can do demodulationwise.

CHETHIK: Well regarding the bandwidth of these optical systems I know that there's an outfit in the San Fernando Valley, the name of which escapes me, where we can buy commercially today a modulator, single mode fiber, and photodetector which has a 10 gigahertz bandwidth. That will run over several tens of thousands of feet, so that is today technology. Just trying to compare this in my mind with what I know about, for example, dual gate switch matrix or switch matrices being developed for various NASA programs like the ACTF. There are 8×8 switches, for example, in existence that are extremely broadband, have very high isolation, low insertion laws, a big dynamic range, etc., a switch in fractions of nanoseconds, well maybe a few tenths of nanoseconds, that don't consume much power, are lightweight, and I'm just wondering if you're addressing the wrong problem. I think there are probably some very excellent applications for optical fiber and switching technology on board spacecraft, but I'm scratching my head about whether onboard switching is something worth putting a whole lot of energy in.

GAGLIARDI: Well I think that is TBD, (to be determined), and hopefully we can get a comment on that. I think that the tens of nanoseconds is about the limit of electronics, and if you want to go into fractions of nanoseconds then optics appear to be the next step here, so we're going to push to that particular level. I'd like to comment, by the way, that in the special issue of *Electronic Switching*, there's an excellent article talking about this switching, and in fact, Ray, it has the a icle from the Columbia people that I referred to. There should be some of their numbers in there, so if you are interested in switching, the kind we are hinting at here, there's a fairly good summary of these papers in this *IEEE Communications Magazine* that just came out recently.

REY: Yes, I would think an application of this would apply to SDI, where you have two to three thousand vehicles you're trying to communicate to from a single satellite. So now you're starting to, well in fact you've gone past the electronic switching technology, and that would be an example.

GAGLIARDI: Right. You are dealing with just these fiber bundles, and that maybe can be extended as well. Okay, I think we should go on ....

DUPREE: Well if you have any further questions, make a note of them as the talks progress and we'll have to go back to the subjects a little bit later. A comment that I have on this whole subject area is that whether optics is a practical alternative to electronics or not, will not be known until we have fully explored the possibilities in optics, and until we have developed our state of knowledge to the point that we're able to make a fair trade. In doing all of this we have to search around for possible applications. In the process of doing the trade studies sometimes we discover directions that we hadn't thought of before. So now I'd like to introduce Bill Steier to move us along a little bit toward more of the bulk optical information processing.

STEIER: This will be a change in pace. My background is not communications; it's strictly optics, so I'll be talking more from the standpoint of where optical signal processing stands. As you know, it's a very big field. There have been many conferences on the topic and all I can do is give you an overview; a feel of where things stand. I'm sure there's going to be many interesting optical signal processing ideas which I'm not going to be able to cover.

The first issue is one we've already heard about: Why do you want to consider optics for signal processing? Why not stay with electronics? Essentially optics has two advantages that are shown in FIGURE 1. It has a high spatial bandwidth and that essentially comes about because you can focus an optical beam very tightly. You can make a pixel size on the order of the focused spot so you have a high two-dimensional bandwidth. The other is that optics is potentially very high speed. We can make optical pulses which are picoseconds or a little less. We can't modulate that fast; the bandwidth of modulation is probably in the 10-30 gigahertz range, but at least we can make pulses that fast. Whether we can manipulate them fast enough is another question.

Most optical signal processing has exploited the high 2-D spatial bandwidth advantage of optics. FIGURE 2 compares optics to electronics in this regard. In any type of signal processing you have two problems: the connectivity between the devices and the nonlinear devices. You need nonlinear devices, such optical switches, modulators, and storage. If you look at optics, you find it's very good at doing the connectivity. It's easy to see that a simple lens can connect together an object and an image with perhaps a million pixels in each. A simple lens does that very easily. But it's difficult or impossible for electronics to wire in that kind of density. That goes back to the fact that photons tend to ignore each other but electrons always have crosstalk. On the other hand you need nonlinear devices, and optics is poor at that. This is also caused by the fact that photons don't like to interact, and electrons interact readily. With electronics it is relatively easy to make nonlinear devices; but with optics it is difficult.

Any kind of optical signal processing has to exploit what optics can do easily, and must live with what optics has difficulty doing. So you try to exploit the high degree of two dimensional connectivity.

You can consider hybrid systems where you use the best of both worlds, and that's the idea of using optical interconnects inside a large silicon computer. Here you let the VLSI do the computing and you wire it together with optics. The problem

with using this hybrid approach in optical signal processing is that the interfaces between optics and electronics become a major problem and will consume so much time that it must be minimized. Hence one always tries, if you're going to use optics, to stay in optics.

The theme of my talk will be the limitations imposed by the optical nonlinear materials and where we stand on optical devices. FIGURE 3 is a rough schematic of a general 2-D optical signal processing system. You must have an input to either turn the electronic signals into optical or to convert an incoherent optical picture input into a coherent optical picture for signal processing. In addition you must have a CPU which defines the type of processing. Examples of CPU functions are Fourier transforms, vector matrix multiplication and matrix matrix multiplication. I'll show some schemes for optically achieving these processes. On the output you will perhaps want to connect the signal back to electronics or perhaps leave it in an optical form. In many schemes you'll want to have some interim optical storage since you may want to store a frame for processing with some earlier frames or later frames.

First of all, let's review some of the input devices. There's really two ways to go, as shown on FIGURE 4. The first uses acousto-optics in which you modulate your information onto an acoustic beam which is launched into an acousto-optic material, typically quartz. The light interacts with the acoustic pattern and converts the electronic information over to the optical beam. These devices are one dimensional with high resolution: on the order of a hundred lines per millimeter.

The second approach is two dimensional. It uses spatial light modulators which are simply two dimensional arrays of optical modulators. There is considerable work in this area because they are a key element in many systems. There are electro-optic approaches, liquid crystals devices, deformable mirrors devices and magneto-optic approaches. The key here is nonlinear optics; you need nonlinear optical materials to do this. FIGURE 5 shows an example of one of them: the liquid crystal device. It's on the market now and Hughes sells one. This device turns incoherent optics into a coherent optical beam. There is an optical photoconductor as shown and a liquid crystal in series across a voltage. Where the write light is intense, it increases the conductivity of the photoconductor, increases the voltage in that region of the liquid crystal and that changes its optical properties, usually by rotating the plane of polarization. The change in polarization is changed into amplitude modulation by an analyzer. Resolutions are in the range of 35 lines per millimeter. Contrast is about 100 to 1, which may be marginal if you're considering a digital system of very high accuracy. They're relatively slow, it takes a couple of milliseconds to turn them on and about 5 milliseconds to turn them off. As Jim pointed out the optical sensitivity of the liquid crystal SLM is about equivalent to enlarging film.

Some of the SLM's being developed are faster, some of them require less light, some of them have slightly better resolution. None of these devices is perfect and there is still a considerable amount of work to be done on 2-D spatial light modulators.

Let me go back now to my general picture again and review some of the transforms you might do by optics. The first one is the familiar Fourier transform, which is useful in many applications. FIGURE 6 shows an approach which uses Fourier

transforms and nonlinear materials to perform a real-time 2-D convolution. Coming in are patterns B1 and B2; both are two dimensional patterns of light that are to be convolved. The inputs are put through a lens that does the 2-D Fourier transform of both. The patterns interact in a nonlinear optical material which has the property that its index of refraction is a function of the intensity of the incident light. This is the four wave mixing idea. These two beams interact and the product is read with an intense plane pump beam. One way of viewing the interacting is that the pump beam and one of the inputs write a hologram which is read with the other input. The product is inverse Fourier transformed which gives B_4 which to a first approximation is the correlation between the two input patterns. Of course this is done at the speed of light, or in reality it is done at the speed of the nonlinear material. How fast the nonlinear material can respond is dependent on the material but it can be in the nanosecond regime.

The problem is always to find a nonlinear material that gives a reasonable amount of energy back in the correlation. You'd like to make the interaction distance long to increase the output power but the longer you make the interaction distance, the poorer the approximation becomes. So the tradeoff is between the power in the correlation beam and the accuracy of the correlation. Using four wave mixing in a nonlinear material, you can get a variety of outputs as shown in FIGURE 7. A and B are two-dimensional information carrying beams and C is a plane wave. Depending on the configuration, you can get out A times the complex conjugate of B or you can get out a multiplication A×B. You can do correlations or convolutions with these approaches.

Let me return to the general picture and discuss some of the vector matrix multiplication ideas and how to do them. The first approach, shown in FIGURE 8, was developed at Stanford; the work of Joe Goodman. It's a relatively simple but very powerful system. The system used a linear array of light emitting diodes. The output of each diode is proportional to the elements of a vector. The output from the upper LED illuminates the first row of the matrix by use of a cylindrical lens which is not shown. Another cylindrical lens collects all of the light from the first column into the first detector. If you follow each beam you can show that this system performs the vector matrix multiplication between the vector which is the intensity of the LEDs, and the matrix which is the transmission of the mask. The system works in incoherent light and can be very fast. Since it works at the speed of light it is equivalent to 10^{12} multiply-add per second. Accuracies of 250 to 1, or 8-bit are considered the limits of this analog scheme. The problem is that the matrix cannot be easily changed since it is on a piece of film. To do this in real time will require one of the spatial light modulators which may have limited speed and limited spatial bandwidth. The approach is limited in two ways: accuracy and the ability to change the matrix.

There are some ways around these limits that people have been working on, which is called the systolic approach. Systolic means that the data is pulsed in a systolic way. FIGURE 9 shows an acousto-optic modulator and a driver. Clocked into the acoustic port in timed sequence are the components of a vector. A linear array of LED's has clocked into them in staggered form all the components of the matrix to be multiplied. When the acoustic wave of

amplitude of x_1 is in front of the first LED, the LED is flashed on with an intensity proportional to the first component of the matrix. The diffracted light beam is proportional to that product. The system keeps pulsing itself as x_1 moves up, and the next matrix component is pulsed in as shown. If you follow through the process you find it essentially does the vector matrix multiplication. With this scheme the problem of updating a two dimensional mask is avoided since everything is driven in time. You have avoided the problem of varying this matrix or varying the vector you wish to multiply. If you consider a gigahertz electro-optic modulator, which is quite reasonable to do; if you consider a 250 LED array, which is tough but possible; this system is capable of about 10^{10} multiply-add per second. The question again gets back to accuracy, because the approach is analog.

People have talked about ways of doing this in a digital format. You can spatially write a number digitally and show that if you spatially convolve two digital numbers, you'll come out with an answer which is the product. The problem is that the answer doesn't come out in a 0-1 format. You must do some electronic processing to turn the answer back into a 0-1 format and that's always the problem. It turns out that this electronic processing to get back to binary is so time consuming, it may not be worth doing optically in the first place. There was a program to develop this commercially which was apparently limited by the electronic processing. I want to expand on the digital approaches to optical signal processing which gets closer to the area called optical computing.

REIFFEN: Can you handle phase as well as amplitude in the mask or in the output of the modulator?

STEIER: Yes ....

REIFFEN: How do you introduce phase in the input or the mask?

STEIER: The usual way to introduce phase is to use the concept that any complex number can be written as three real numbers. You give me the component at 120°, at 240° and 360°. So any complex number can be represented by three positive numbers, and therefore you do the processing essentially three times, to get the complex answer. You always work with positive real numbers, but you do it redundantly.

REIFFEN: But then, how do you then reconstitute the optical image, which involves an honest-to-God, let's say, phase-sensitive operation in terms of what you're trying to simulate?

STEIER: I think, as I understand it, that the approaches to this have been to get a complex array of numbers which you break up into three positive arrays of numbers, and always stay in that format. In other words you don't optically ever attempt to go back to a complex optical pattern. You stay in these three incoherent positive number patterns for your processing. The idea being just to be able to process complex numbers, not to make a complex picture, or to make a picture with phase in it.

REIFFEN: Well, I understand what you're saying but that may not cover all applications of interest.

STEIER: Yes, you're right.

DUPREE: You have a legitimate concern about the question of the electronic overhead and preparing the three numbers, three sets of numbers at the input and then decoding in the output. Something to bear in mind in all of these processing schemes,

you'd like to do as much of the processing as possible in the optical domain. Somehow or other we have to get away from that electronic optical interface. That's one of the big problems that we're trying to overcome.

STEIER: FIGURE 10 is an example of digital optical computing or optical signal processing, where you try to get the advantages of digital. Shown is a two dimensional array of nonlinear optical elements. In this case it is a two dimensional array of NOR gates. The two inputs to the NOR gate come in from the left side, the output is a beam of light that emerges on the right side. The system is wired together with a complex hologram. It mimicks a conventional digital optical circuit; the circuitry is the hologram; the NOR gates are in the array, and you can mimick whatever optical circuit you desire by the hologram you use. It has the advantage of parallelism. The computation goes to some extent in parallel. You can have parallel input, parallel output. It avoids some of the pin-in pin-out constraints of very large VLSI. Again the problem is the array of NOR gates, and there you're limited by nonlinear optics. A lot of the work is currently underway trying to make these optical nonlinear gates. The realization of an optical digital computer is limited by the nonlinear gates and much work has to be in the devices.

Let me just talk for a minute about the switching requirements in the 2-D transforms Bob Gagliardi mentioned the switching needs. FIGURE 11 shows one example of the state-of-the-art. There's a lot of work on photonic switching as evidenced by a meeting in Tahoe in March reviewing this field. Probably the most advanced technology is the lithium niobate waveguide switches shown in FIGURE 11. This device is fabricated in integrated optics. There are techniques for writing waveguides in the block of lithium niobate. As shown there are two waveguides which are close enough together for the evenescent mode fields to overlap and interact. This is a directional coupler in which two waveguides are coupled together by the evenescent field. The lithium niobate is electro-optic and there are four electrodes deposited near the waveguides which are run in push-pull. Depending on the voltage that is applied, the index of refraction in the region between them is changed and the coupling between the two waveguides is changed. These devices usually require with a few volts; perhaps 5-10 volts. You can switch the light beam here to output 4 or vice versa you can switch input 2 over to output 3. This is a voltage controlled optical switch. Significant progress has been made at AT&T Bell Labs; they've made an 8×8 array. It's an optical crossbar: 8 in, 8 out. It also has the broadcast capability: one input can talk to all of the outputs. Gigabit switching rates and low crosstalk are claimed.

As the last item I will review the prospects for 2-D optical storage. That gets back to real time holography again. FIGURE 12 shows a nonlinear material and a pattern to be stored in memory. The pattern is complex and must be stored in amplitude and phase. The reference beam and the pattern write a hologram in the nonlinear material. This is a material whose index is a function of optical intensity. The pattern is stored until read with a reference beam to retrieve the information. The materials currently of the most interest are the photorefractive materials. Storage times in these materials can be from milliseconds up to hours in the dark. Write times are from microseconds to minutes

WHY OPTICS?

* **High Spatial Bandwidth**

  pixel ~ $4\lambda^2$

* **High Speed**

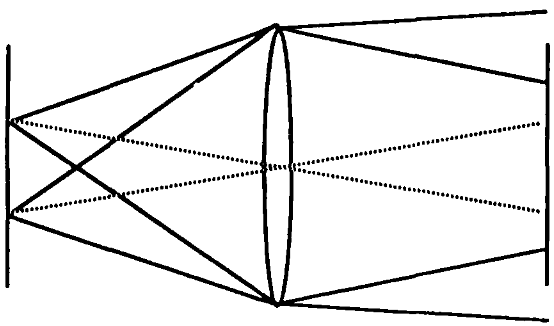
  10^{-12} sec pulses

Figure 1

FIGURE 2: Comparison of optics and electronics in the two requisites for optical signal processing: connectivity of the elements and nonlinearity of the elements.

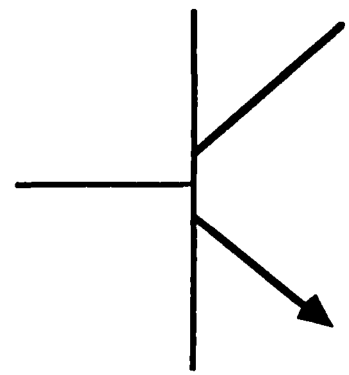
	Connectivity	Nonlinearity
Optics	+	−
Electronics	−	+

Connectivity



* Optic Fourier Transform
* Dense 2-D Wiring

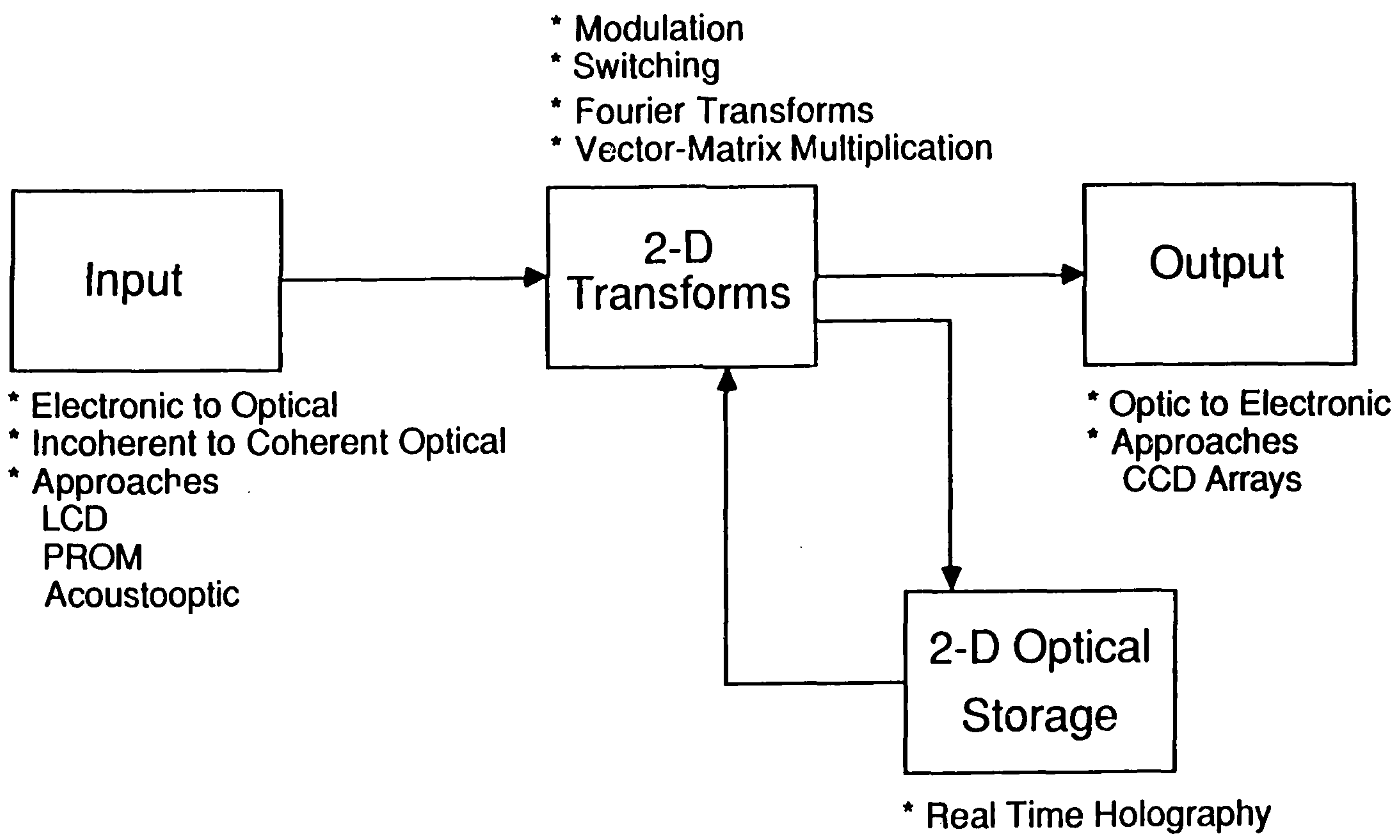
Nonlinearity



* Switching
* Modulation
* Storage

FIGURE 3

2-D Optical Signal Processing

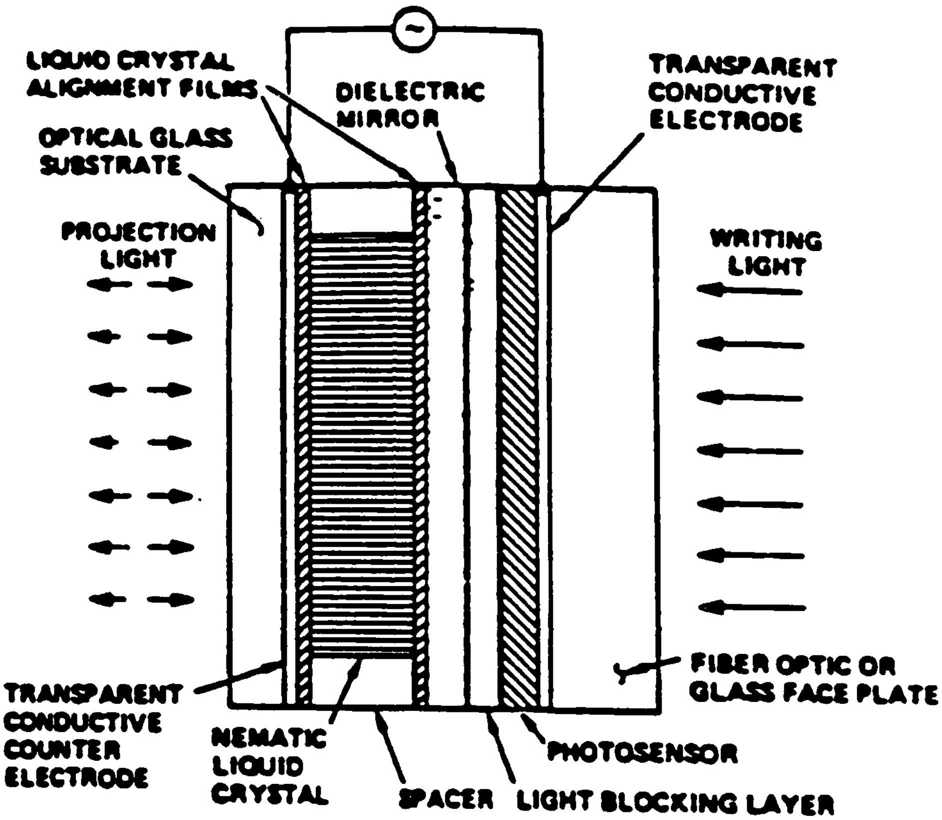


INPUT

* Acousto-optic Devices
  - One dimensional ~100 lines/mm
  - Electronic/optic
* Spatial Light Modulators
  - Two dimensional
  - Electro-optic
  - Liquid crystal
  - Deformable Mirrors
  - Magnetrooptic
  - Optic/optics, electronic/optic

FIGURE 4: Summary of devices for inputing data into optical signal processing systems.

LIQUID CRYSTAL LIGHT VALVE

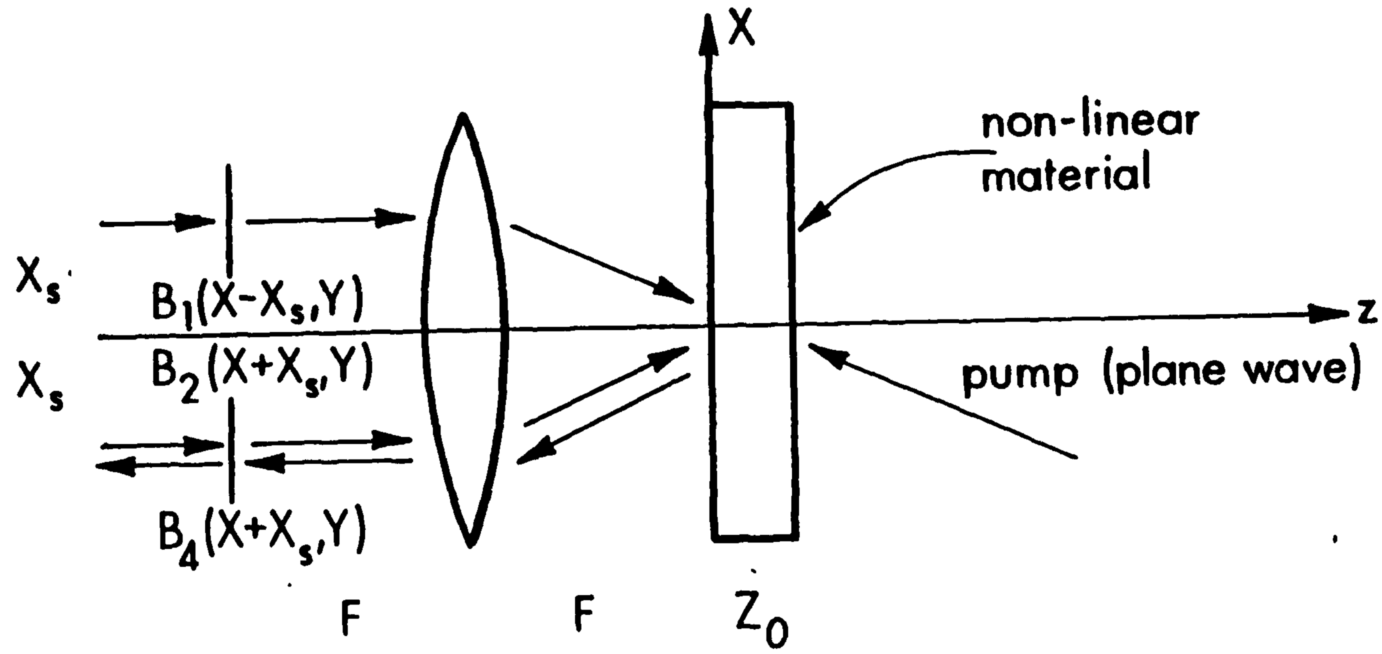


Hughes Si LCLV

Aperture	40 mm
Resolution	35/mm
Contrast	100:1
Response Time	2/5 msec

FIGURE 5: The Hughes Liquid Crystal Light Valve, Hughes Industrial Products Division, Carlsbad, CA.

- Correlation via non-collinear, degenerate FWM:



$$B_4(X,Y) \sim B_1(X,Y) * B_2^*(X,Y)$$

FIGURE 6

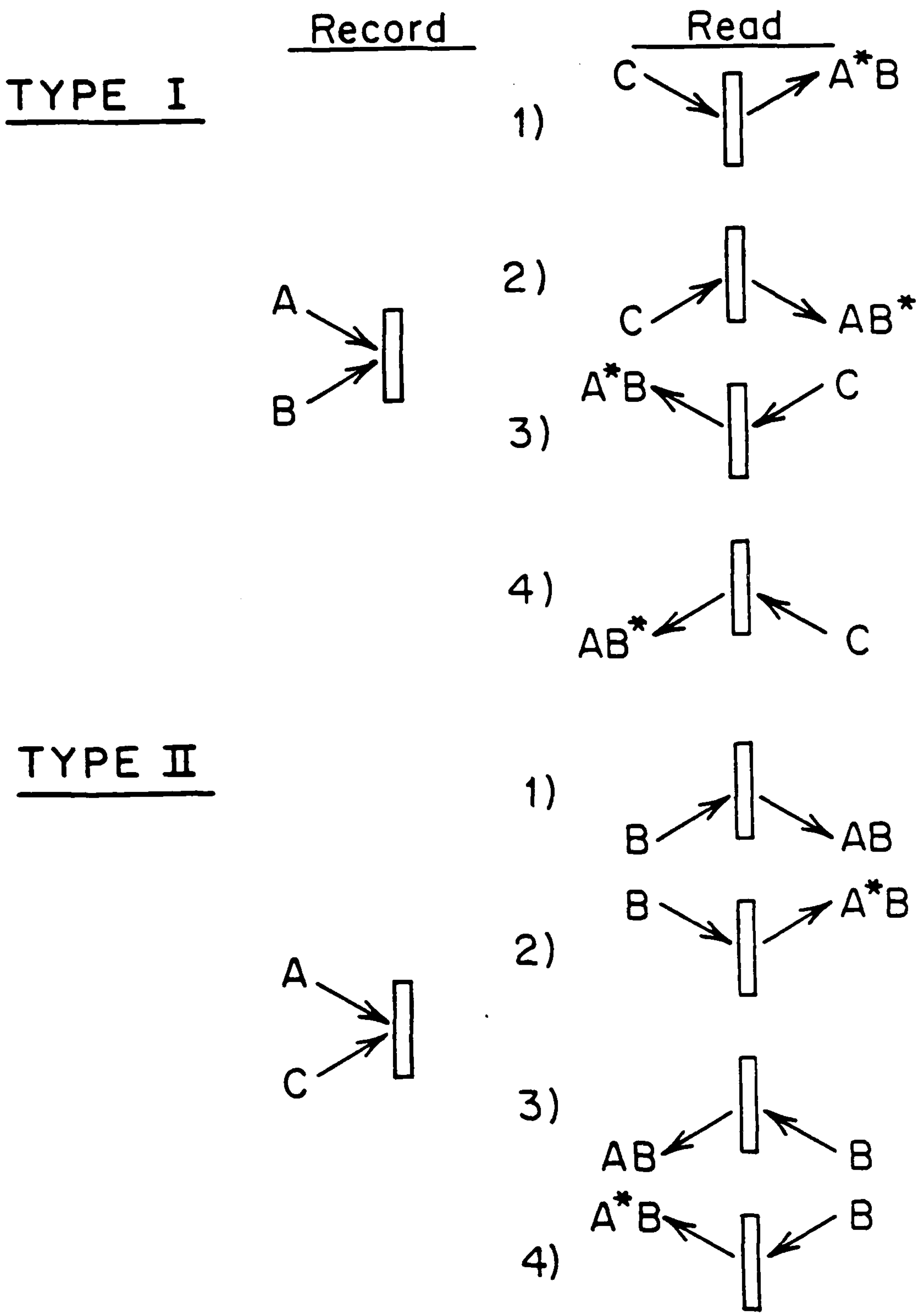
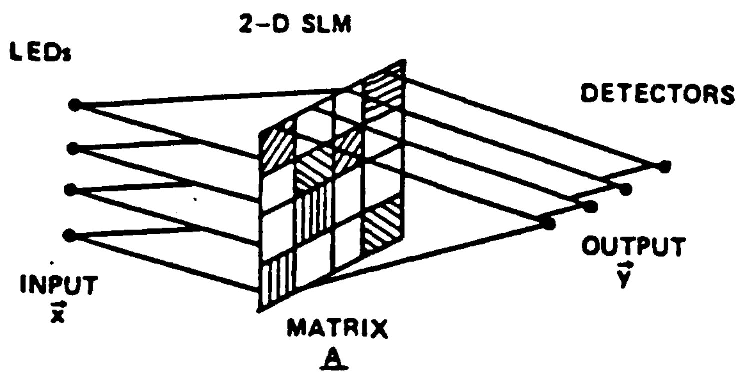


FIGURE 7: Products possible using four wave mixing in a nonlinear material. A and B are data carrying input patterns; C is a plane wave.

VECTOR MATRIX MULTIPLIER

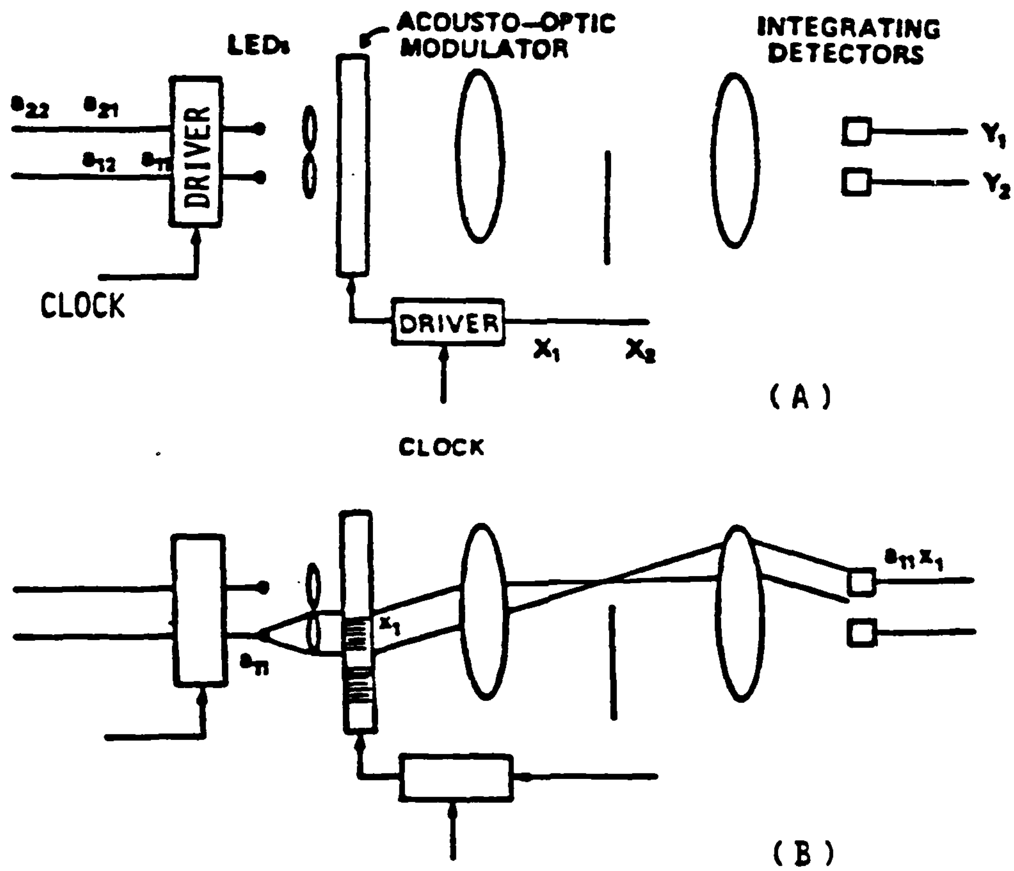


- Incoherent Light
- Fast/Parallel - 10^{12} multi-adds/sec
- Accuracy 250:1 (8 bit)

FIGURE 8: The Stanford Vector Matrix Multiplier. Reference: J. W. Goodman, A. R. Dias, and L. M. Woody, Opt. Lett., 2, 1 (1978).

FIGURE 9: Systolic vector matrix multiplier. Reference: H.J. Caulfield and W.T. Rhodes, Opt. Computing, SPIE 456, 2-14

SYSTOLIC ACOUSTO-OPTIC VECTOR MATRIX MULTIPLIER



Projected Capacity

* 1 GHz AO Modulator
* 250 LED Array

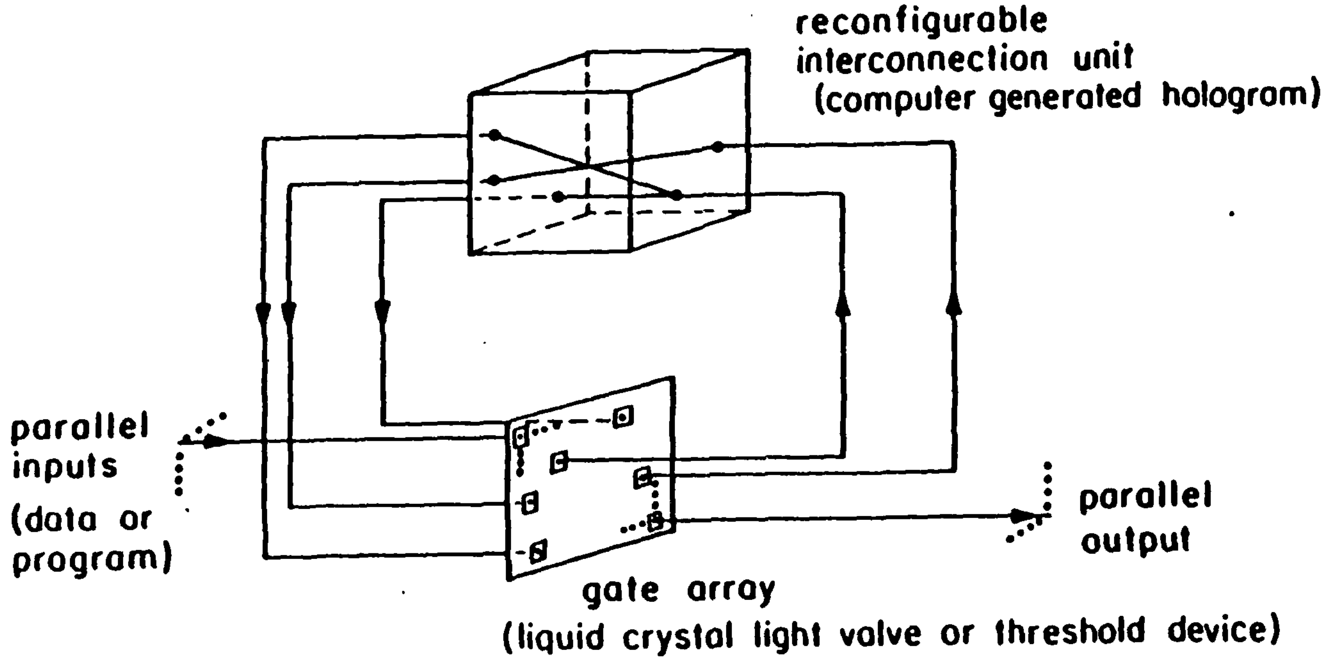
6 x 10^{10} mult-add per sec

ACCURACY? HARDWARE?

FIGURE 10: Digital Optical Logic System. Reference: B.K. Jenkins, et al., Appl. Opt., 23, 3455 (1984).

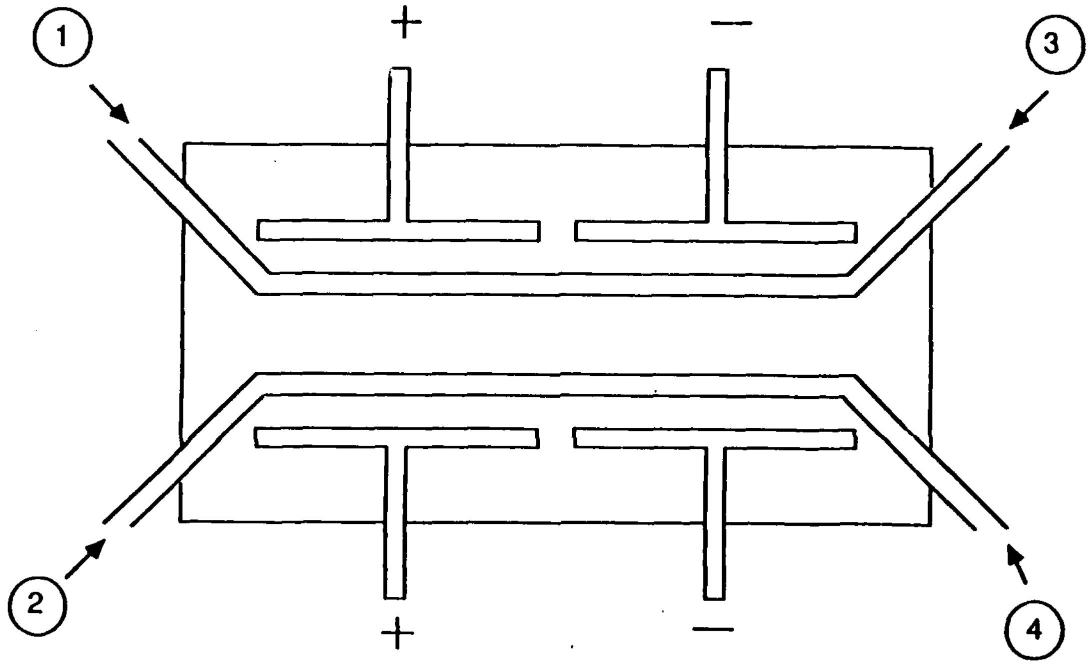
DIGITAL OPTICAL SEQUENTIAL LOGIC SYSTEM

A.A. Sawchuk
B.K. Jenkins



- BINARY COMPUTER - NOT VON NEUMANN ARCHITECTURE - DIGITAL ACCURACY AND DYNAMIC RANGE
- PARALLEL I/O POSSIBLE - NO PIN-IN/PIN-OUT CONSTRAINTS
- ARBITRARY INTERCONNECTIONS - GLOBAL/LOCAL

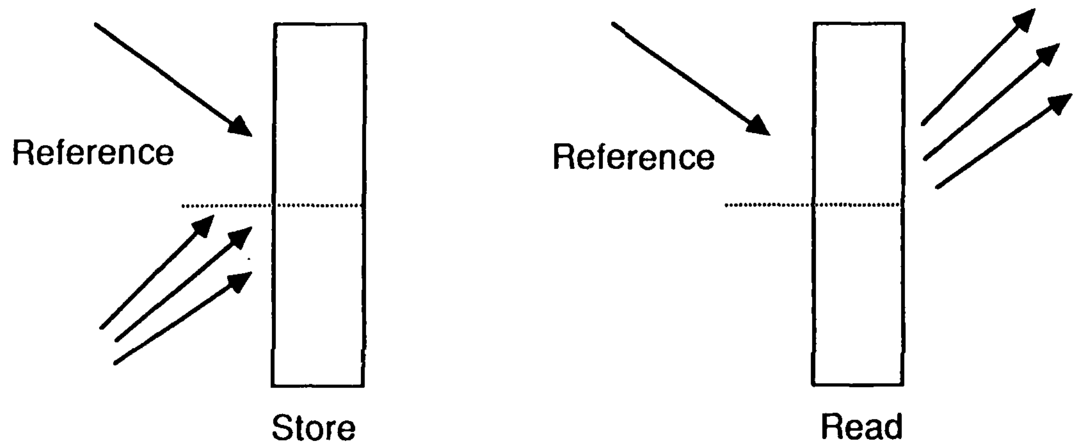
Ti:LiNb0 $_3$ Waveguide Switch



* 8x8 nonblocking switch array
* Gigabit switching rates
* Low crosstalk

FIGURE 11

2-D OPTICAL STORAGE
REAL TIME HOLOGRAPHY IN PHOTOREFRACTIVE MATERIALS



* Storage Time - milliseconds to hours
* Write time - microseconds to minutes
* Resolution - 100/mm
* Efficiency is inversely proportional to speed

FIGURE 12

COMPARISON OF PHOTOREFRACTIVE MATERIALS

	Wavelength Range	τ_d	n^3r(pm/v)	Sensitivity (cm^3/Kjoule)	Typical Δn_{ss}
GaAs:Cr	0.8 - 1.8μm	10^{-4}s	43	30	$5x10^{-6}$
InP:Fe	0.85 - 1.3μm	10^{-4}s	52	30	$5x10^{-6}$
BaT_iO_3	0.4 - 0.9μm	10^2s	11300	5	10^{-3}

FIGURE 13

with fairly high resolutions on the order of a hundred lines per millimeter. In all of these materials the efficiency of the scattered out beam is inversely proportional to the write time. It is a compromise between the intensity of the output and the time to store the information.

REIFFEN: Does that reference have to be coherent?

STEIER: Yes, right. These two beams have to be coherent to one another to write a hologram.

FIGURE 13 shows some properties of selected photorefractive materials. Chrome-doped gallium arsenide and iron-doped indium phosphide are used in the infra-red where laser diodes are available. Barium titanate is an often used material for phase-conjugation which is used in the visible. The parameter Δn_{ss} gives the index change that can be created in these materials. It gives an idea of the efficiency of the hologram that can be created. Holograms in barium titanate can be three orders of magnitude more efficient than in gallium arsenide or indium phosphide. The limitation in barium titanate is the dark relaxation time which is the time required to redistribute electric charge in the material. It will require seconds at perhaps tens of seconds to write a hologram in barium titanate but the hologram is efficient. In gallium arsenide, holograms can be written in microseconds but they are not nearly as efficient. However, GaAs and InP are the best materials available for microseconds or perhaps nanosecond writing of holograms.

DUPREE: Can you change the index by changing the intensity of illumination?

STEIER: Yes, this is the maximum value. The index change will saturate at that value. Well, maybe this is a good spot for me to stop. Optical signal processing is still in a state of evolution with many systems concepts awaiting devices. The device technology still has a way to go before the devices are reliable enough, and work well enough to exploit the potential advantages of optics. Perhaps that's the same situation in communication systems. As the devices get more reliable, the systems engineer will feel much more at-home in trying to use them in a system. It's perhaps difficult to consider undeveloped devices. You don't want to gamble on those when you're talking about a system.

DUPREE: Are there any further questions?

WEBER: I have a very general question, that is, to what extent do you think that applications, say from communications or control people, could indeed drive where your technology goes, as opposed to you finding a material somehow, and that makes the device, and you have no idea where it's going.

STEIER: I agree it's a two-way street, but it's not a matter that we don't want better materials. Certainly there are a lot of applications around that could be exploited. But certainly the kind of devices that would be pushed should be influenced by the systems side. And that's also true in the optical computing field. Devices that are pushed should be influenced by the computing schemes that people are talking about. And also, it goes without saying that the systems needs will perhaps produce the money to drive the device research.

REIFFEN: A comment... We've observed that in many of the signal processing applications, one of the big problems is the interface between the electronics and the optics. Perhaps the most fruitful area for optical computing is in applications where the input to be processed is in fact

optical, is in fact a picture or an image of some kind, thereby eliminating that need for that interface in some sense.

STEIER: You are absolutely right and it may be that the role of optical communicating will be in what is called these ill-defined problems where neural net-type computers are appropriate.

DUPREE: I'll comment on that again. You need to get the sensitivity and the resolution in the same neighborhood as of the film in order to establish these kinds of applications. There simply aren't enough protons in real-world scenes to act efficiently on the specialized modulators in real time. If you are going to make very long exposures, then we could certainly work with available materials.

SULLIVAN: I would summarize this presentation as the communication systems (straight-down-the-middle) approach to optical processing. [CHART #35] entitled "Main Contribution" summarizes what I have to say. Other material is in this package that I won't present. I am presenting an architectural approach to the use of optical processing in electronic systems. I consider it to be an initial effort on overall communications systems implemented with optical processing. Linear and non-linear techniques are employed. Non-linear optics are necessary for communications because, in fact, most of the processing in communications is non-linear. Linear processes are straight forward. The extrapolations to linear techniques in optics are almost obvious. For communications the need is to do devices like modulators, demodulators, mixers, filters, matched filters; those kinds of implementations that we use in electronic systems; and do them optically. The goal is to avoid moving from optics to electronics and back; to perform all of the processing operations on light beams using optics. I'll show an overall diagram which will provide guidance in terms of optical device research. I will not comment much about what those devices can do. I think Bill Steier has done an excellent job on that. I will show how I can use various kinds of devices. Th communications satellite transponder is what I've emphasized. I consider it a major contribution. But the long term goal is to have a bulk piece of material as one has a chip in electronics. Use of the three dimensions that are available inside that bulk material will create whatever devices and phenomenon are needed. The integrated optics bulk crystal can be a big deal in the future. The potential of the technology is outstanding.

[CHART #2] shows a microwave communications transponder. There is considerable microwave channelization in this transponder. The input and output circuits could be monolithic microwave up to very high frequency, even 50 GHz. In the optical implementation of the microwave transponder shown in [CHART #3 and 4] you receive microwave, you transmit microwave, and everything in between is in the optics. It is microwave channelization of multiple beams, a microwave switching and routing, then demodulation and demultiplexing to separate it into various channels, baseband switching and routing, then multiplexing and modulation onto microwave carriers, followed by microwave power amplifiers and radiating elements (antennas). The optical implementation of the transponder is on two charts. Part A of [CHART #3] depicts the input side of the transponder. The antenna elements are followed by microwave pre-amplifiers. The individual channel time signals are converted into spatial equivalents using acousto-optic cells. A lens is employed to

perform a Fourier transform. Along a spatial dimension at the back focal plane of the lens is the spectrum of the one antenna's output. Along the vertical dimension is the outputs of the various input antennas. And then there is an interconnection network, implemented optically, to switch and route the microwave carriers. On the input side of the Part B Block Diagram of [CHART #4] is shown an optical implementation of the microwave demodulation, possibly within a bulk device. The optical interconnection networks can be implemented in various ways, perhaps employing holographic elements. The baseband (or video) waveforms from the interconnection and multiplexing network will be modulated onto microwave carriers (still in the optical domain), then converted back to microwave by optical detection. The resulting microwave carriers are amplified and transmitted by an antenna configuration.

[CHART #5] depicts the conventional configuration (electronic implementation) of an optimal receiver for a pulsed microwave signal of unknown amplitude, phase frequency, pulse width and time-of-arrival. It assumes a maximum likelihood detection algorithm. An optimal processing realization of this receiver is shown in a later chart. Note that the operations shown here are all linear except for the square law (envelope) detector. This is an important consideration in the equivalent optical implementation. Most of the interesting techniques described in this presentation are nonlinear with various waveforms being multiplied together.

The first step in the optical processing of a microwave signal is the conversion of the microwave time waveform to an optical spatial waveform or variation. Devices that accomplish this are called spatial light modulators. The emphasis in this development has been on acousto-optic devices since the acoustic velocity of propagation in the transverse spatial plane (very slow - 10^4 meters/sec) provides a reasonably small spatial aperture. The modulation format chosen is double sideband amplitude modulation of the microwave signal as a spatial variation on the optical carrier. [CHART #6] defines the assumed spatial utilized throughout this development. It should be noted that the spatial interference pattern generated from the microwave signal can be observed on the light beam (the intensity would be noticable if visible light is used and it is focused on a screen). As is pointed out on the chart the major limitation on the signal operations that can be performed is the observation time bounds that result from the aperature limit corresponding to the transverse length of the acousto-optic cell. [CHART #7] depicts the hypothetical spatial light modulator of the above assumptions, showing the signal spatial variation on the optical carrier. A mathematical breakdown of this waveform into its constituent microwave, optical and spatial components is given in [CHART #9]. A single frequency microwave carrier produces three plane waves as the modulated optical beam.

Within the goal of accomplishing all signal processing operations on optical beams without coversion back to electronics, it is necessary to have devices that allow one optical beam to operate on another optical beam in prescribed ways. The major ground rules for such hypothetical optical devices, as summarized in [CHART #10] are coherence (relative and absolute) as generated from a single laser source, uniform transverse velocity and wavelength, and possessing the define beam-to-beam complex envelope interactions. The specific input/output relationships

for the hypothesized optical devices are given in four cases of [CHART #11]: Amplitude Product Device, Phase Product Device, Frequency Product Device, and Squared Amplitude Device. As will be shown later, these four devices and the assumed Spatial Light Modulator allow the configuration of all of the desired analog signal processing operations to effect an all optical synthesis of the communications and detection systems of interest.

As a first example of all optical realization of electronic signal processing operations, [CHART #12] depicts an amplitude modulator which is implemented by conversion of the baseband and carrier functions to optical spatial variations of the form assumed and a multiplication of the two complex envelopes in the amplitude product device, followed by bandpass filtering and conversion back to an electronic (or electrical) output. This is very much more complicated than a conventional AM modulator implemented in electronic form with simple diode functions, but does provide an existence proof of optics as an alternative approach.

In this same fashion a phase modulator has been configured from two phase product devices, spatial filters about the upper and lower spatial sidebands and the appropriate beam splitter and summer. This is shown in [CHART #13] as operations on optical complex envelopes to illustrate the all optical realization of a complex signal operation. A related, but distinct, similar device synthesis is represented in [CHART #14]. This Voltage Controlled Oscillator (or Frequency Modulator) is a critical component of the later derived phase locked loop. As is true in electronics the operations of phase and frequency modulation are very distinct in their optical realizations, although they are analogous and related.

A particularly interesting electronic receiver function for implementation with optical techniques is the FM discriminator, which performs a noncoherent demodulation of a frequency modulated microwave carrier. The conventional electronic formulation is shown in [CHART #15], wherein the FM detection is accomplished in two steps -- first the conversion from FM to AM; then a noncoherent or envelope detection of the AM. The FM/AM conversion is effected by slope detection on a linear amplitude versus frequency portion of a filter. The same approach can be employed for an optical realization; Fourier transform to the frequency domain; a linear transmittance versus spatial frequency optical function; followed by amplitude detection. An example of the Fourier transform operation on an optical spatial variation of three microwave carriers is shown in [CHART #16], for the two extreme cases of a large relative aperature limit and a small relative aperature limit. The complete optical implementation is shown in [CHART #17], with particular emphasis on the bandpass slope detection transmittance function in the Fourier transform plane. The optical component synthesis of this discrimination is detailed in a later chart.

Pervasive throughout analog processing of electronic signals is the phase locked loop (PLL). For optical processing if the equivalence to this device can be developed, many avenues can be opened. The basic functional PLL is shown in [CHART #18]. As before, all signals are assumed to be of the prescribed format-spatial double sideband, less than 100% amplitude modulated onto the opical carrier. Each of the basic component parts must be realized in an optical form that operates on the complex envelope of the optical waveform: the bandpass filter, the phase

detector, the loop filter and the voltage controlled oscillator. [CHART #19] depicts the envelope waveforms at important points within this all optical phase locked loop. The loop into and the VCO waveforms are shown in an approximated locked condition which corresponds to nearly 90° phase shift of the microwave carriers. The filtered output of the phase detector is also shown; it is the error signal which drives the overall loop toward lock. The nature of the total phase detector output is determined by the operation of the loop, i.e., whether it is operating in a modulation tracking on modulation restrictive mode.

An RF mixer, a basic electronic component for microwave downconversion or upconversion, as implemented optically, is diagramed in [CHART #20]. The amplitude product device multiplies the two complex envelopes, thereby creating sum and difference frequencies of the microwave carriers. The spatial bandpass filter, which operates in the Fourier transform plane, selects the difference (or sum) portion of the output, thereby accomplishing the downconversion (or upconversion).

Now it has been demonstrated how a major set of the desired electronic processing operations can be accomplished with the defined hypothetical spatial light modulator and hypothetical optical devices. The potential hardware implementations of these components must also be addressed. Because of the assumed spatial waveform of double sideband amplitude modulation the acousto-optic modulation cannot be effected by the conventional Bragg cell modulator. [CHART #21] outlines three approaches to the desired optical beam modulation. Two Bragg cells with the carrier component and modulation level appropriately set can synthesize the desired modulation format. The formulation is shown in [CHART #23]. Each Bragg cell generated one sideband and a carrier component. The cell input prisms orient the beam at the Bragg angle; the output prisms align the carrier components for coherent combining (microwave phase as well as optical phase). With the modulation levels correctly set this arrangement can accomplish the desired microwave modulation of the optical carrier. The other candidate techniques of [CHART #21] utilize acousto-optic cells operating in the Raman-Nath mode of optical carrier phase modulation.

If this phase modulation index is constrained to a reasonable level the first sideband level relative to the carrier component can be controlled to approximate amplitude modulation of the spatial carrier, provided the phase shift of the first sidebands is corrected. This technique, depicted in [CHART #22], is the operating principle of the well known Zernicke phase contract microscope. A mathematical model of the Bragg cell acousto-optic modulator, from which its operation can be analyzed, is shown in [CHART #24]. The amplitude level of the various spatial frequency components are available from the literature and are fundamental to the phase match operation that is employed. The third technique of spatial light modulation is depicted in [CHART #25]. In this case the Raman-Nath A-O cell phase modulates the optical carrier to a level consistent with high efficiency, thereby generating a family of spatial sidebands. The discriminator function described previously is used to convert this phase modulation to amplitude modulation to effect an overall spatial light modulator that is theoretically linear. This modulation process has the potential to produce a device that has very high dynamic range for multiple carrier microwave signals.

The limitation on signal processing operations due the A-O cell length (translated to signal time aperture) has been pointed out. Since this constraint on signal frequency resolution is a major concern, techniques for increasing this signal observation time are of interest. One promising scheme is a raster scan of A-O cells in series as [CHART #26] demonstrates. This method has been implemented in hardware, resulting in a two dimensional pattern of coarse frequency resolution in one dimension and fine frequency resolution in the orthogonal dimension. The second coordinate can be viewed as a linear sampling of the input waveform, thereby able to accomplish fine frequency representation within the constraints of the sampling operation. A raster scan consisting of many (100 or more) A-O cells of the highly linear nature described previously has the potential for providing the spatial modulation function desired for the analog optical processors of interest.

The hypothetical optical-to-optical devices of [CHART #11] can be synthesized with two basic components. One of these is a phtorefractive crystal that produces a spatial derivative of the refractive index proportional to changes of intensity of the modulating beam. This is analogous to the A-O cell in the modification of the crystal refractive index but in this case is responsive to a light beam. This basic component can be employed to formulate all four hypothetical optical devices. A second attractive optical component is the phase conjugate mirror realized in the four wave mixing configuration. This conjugator can provide direct beam complex amplitude multiplications for amplitude product device and squared amplitude product device. [CHART #27] summarizes these device realizations. [CHART #28] illustrates the geometry of the four wave mixing phase conjugator, the mathematical representation of which is given in [CHART #29]. To provide this conjugation function the basic crystal is assumed to have a third order nonlinearity of sufficient level for the appropriate output amplitudes to be generated.

The Foster-Seeley discriminator of [CHART #17], implemented in all optical components, including the phase conjugate mirror as an AM detector, is depicted in [CHART #30]. For this implementation it is assumed that the microwave carrier is FDM/FM, i.e., it is multiple channel with the individual channels frequency multiplexed in the baseband. A detector array in the Fourier transform plane of the output is able to perform activity detection of the channels. In addition, with an implementation of demultiplexer of [CHART #31], the individual channels are readily separated, thereby completing a demodulation/demultiplexing operation.

The realization of the phase locked loop of [CHART #18], utilizing all optical components as discussed, is shown in [CHART #32]. The input is shown as a microwave carrier which is spatially modulated onto the optical beam by an acousto-optic modulator of the type previously discussed. All of the various optical signals are derived from the same laser source, allowing the use of coherent processing technologies throughout. The modulated input could also be the result of prior operations on optical beams, therefore already existing in the proper optical format. The bandpass filter is implemented by an aperture limit in the Fourier plane. The phase detector is one of the previously defined amplitude product devices, here implemented with a 3rd-order nonlinearity such as the phase conjugate mirror. The loop filter is assumed to be synthesized as a

TRW

Analog Optical Processing of Radio Frequency Signals

Dan Sullivan

May, 1987

CHART #1

TRW

Demodulation/Remodulation Satellite Microwave Transponder

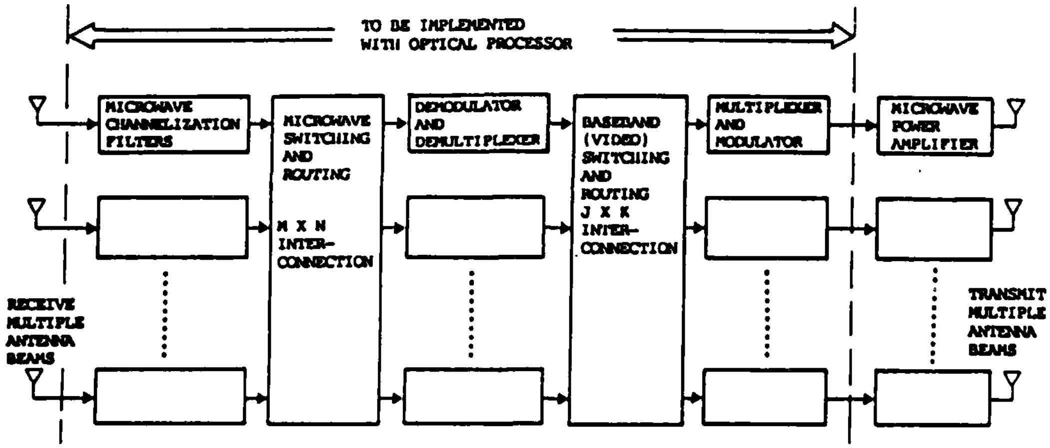


CHART #2

TRW

Proposed Optical Implementation of Satellite Microwave Transponder (Part A)

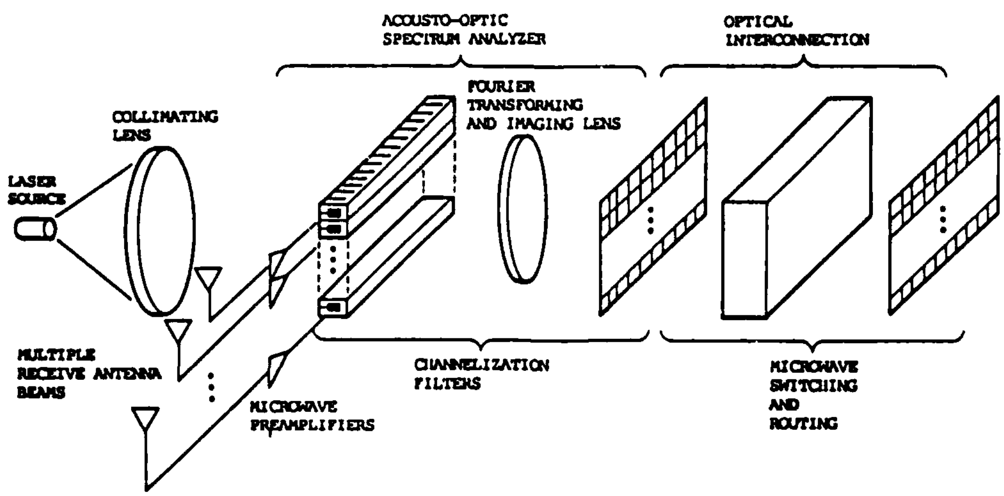


CHART #3

Proposed Optical Implementation of Satellite Microwave Transponder (Part B)

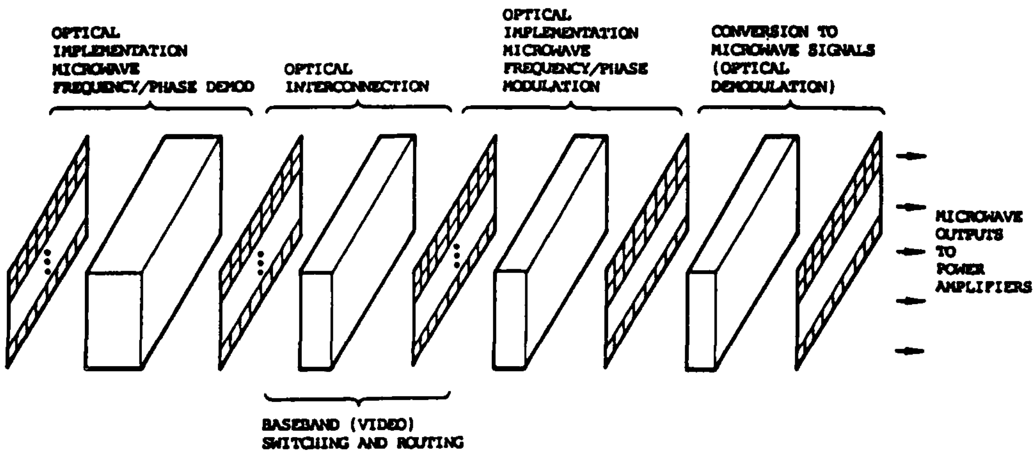
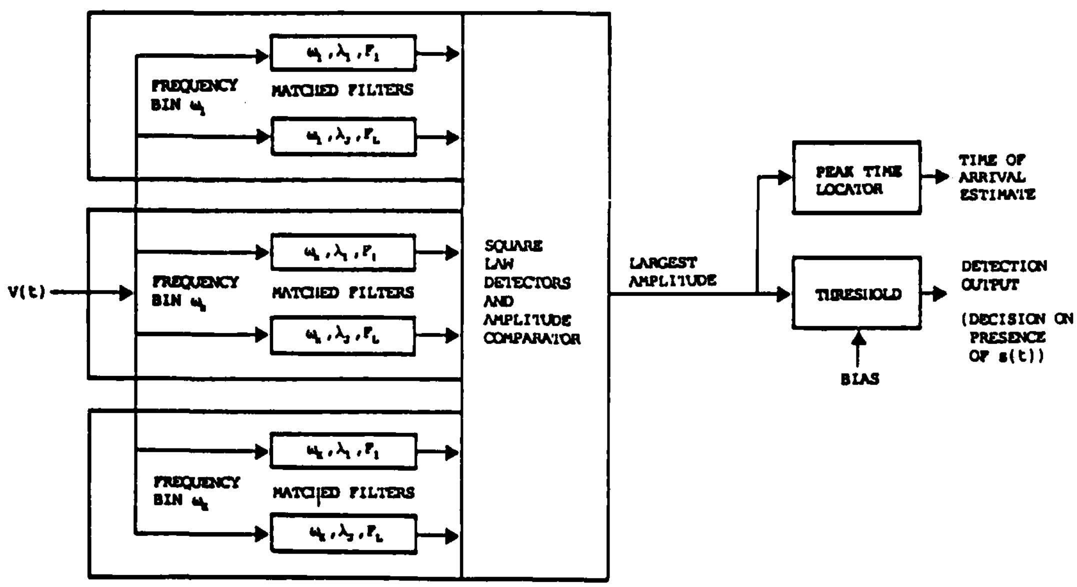


CHART #14

Electronic Implementation



ω_k = radian carrier center frequency of k^{th} filter bin

λ_j = pulse width of j^{th} matched filter function

F_l = PRF (pulse repetition frequency) of l^{th} matched filter function

CHART #15

TRW

Conversion From Microwave To Optical

- Emphasis On Acousto-optic Devices To Provide The Transverse Propagation Velocity And Wavelength Commensurate with Optical Beam. Microwave Modulates Refractive Index

- Utilized Amplitude-Only Modulation Of Interference Pattern Through Double Sideband Spatial Amplitude Modulation

$$B_0(x,y,z,t) = A_0\,\delta(y)\,\mathrm{rect}\left(\frac{x}{s}\right)\left[1 + m_v v\left(t - \frac{x}{s}\right)\right] \cos\left[2\pi\nu t + \phi(x,y,z)\right]$$

- Major Limitation Is Aperature Limit W Which Corresponds To Observation Time $\frac{W}{s} = T$ and Frequency Resolution $\frac{1}{W}$

CHART #6

TRW

Hypothetical Spatial Light Modulator

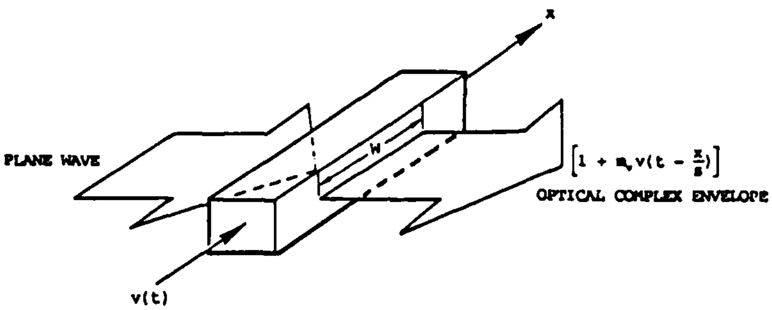


CHART #7

Signal V(t) Converted to Optical Spatial Variation

TRW

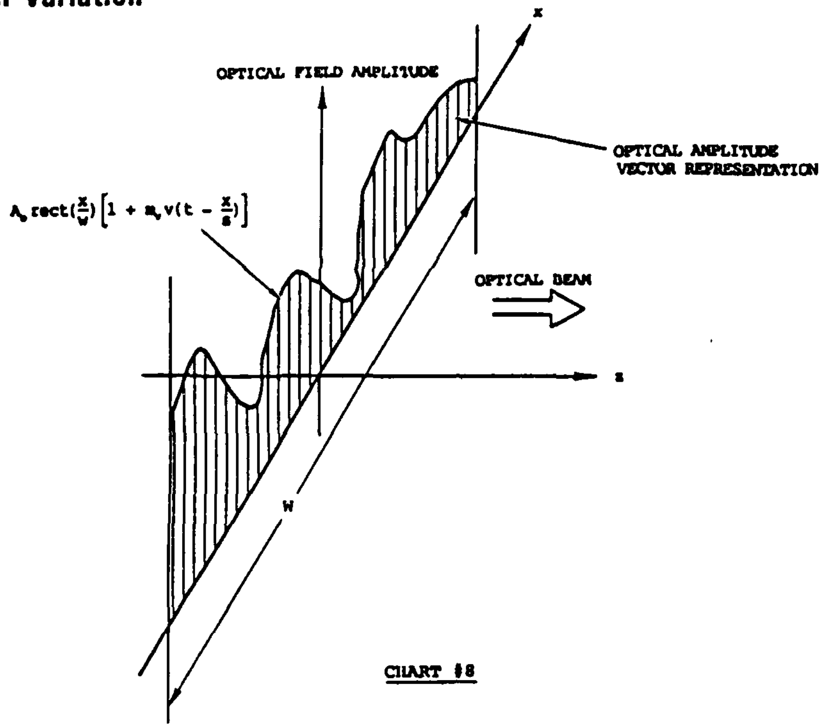


CHART #8

TRW

Analysis Of Spatial Modulated Optical Beam

$$B_0(x,y,z,t) = \mathrm{Re}\{A_0\,\delta(y)\,\mathrm{rect}(\tfrac{x}{W})\exp\left[-j2\pi\nu t - j\phi(x,y,z)\right]$$

$$+m_v \sum_{i=1}^{N} \mathrm{Re}\{A_0\,\delta(y)\,\mathrm{rect}(\tfrac{x}{W})\underbrace{A_i a_i(t - \tfrac{x}{s})}_{\text{MICROWAVE AMPLITUDE MODULATION}}\exp\left[\underbrace{j2\pi(\nu - f_i)t}_{\text{DOPPLERSHIFTED OPTICAL CARRIER}} + \underbrace{j\phi(t - \tfrac{x}{s})}_{\text{MICROWAVE PHASE MODULATION}} \underbrace{- j\frac{2\pi f_i x}{s} - j\phi(x,y,z)}_{\text{SPATIAL PROPAGATION TERMS}}\right]\}$$

$$+m_v \sum_{i=1}^{N} \mathrm{Re}\{A_0\,\delta(y)\,\mathrm{rect}(\tfrac{x}{W})\overbrace{A_i a_i(t - \tfrac{x}{s})}\exp\left[\overbrace{-j2\pi(\nu + f_i)t}\overbrace{- j\phi(t - \tfrac{x}{s})}\overbrace{+ j\frac{2\pi f_i x}{s} - j\phi(x,y,z)}\right]\}$$

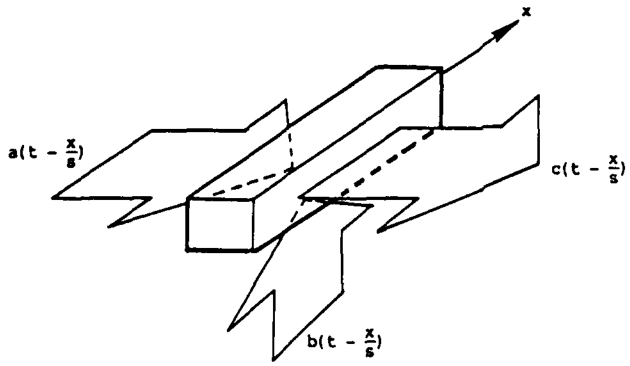
CHART #9

Hypothetical Optical Devices

- Operate On One Optical Beam With Other Optical Beams.
- All Beams Assumed To Be Generated From Single Coherent Source (Laser). Identical Phase and Wavelength
- Must Maintain Transverse Velocity And Wavelength On Output Beam.
- Defined In Terms Of Interaction Among Optical Complex Envelopes.

CHART #10

Hypothetical Optical Devices for Optical Domain Signal Processing



RELATIONSHIP OF OPTICAL BEAM COMPLEX ENVELOPES

Case 1: $c(t - \frac{x}{s}) = a(t - \frac{x}{s})b(t - \frac{x}{s})$: Amplitude Product Device

Case 2: $c(t - \frac{x}{s}) = a(t - \frac{x}{s}) \exp\left[jk_p b(t - \frac{x}{s})\right]$: Phase Product Device

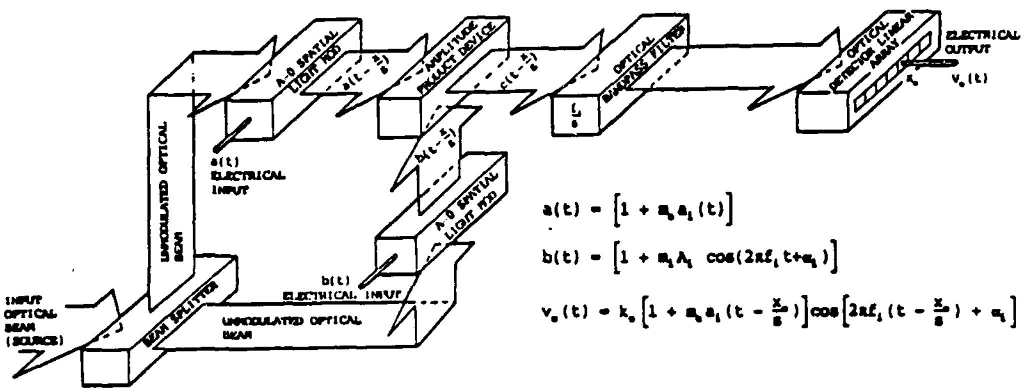
Case 3: $c(t - \frac{x}{s}) = a(t - \frac{x}{s}) \exp\left[jk_f \int_0^x b(t - \frac{\alpha}{s})d\alpha\right]$: Frequency Product Device

Case 4: $c(t - \frac{x}{s}) = a(t - \frac{x}{s}) \left[b(t - \frac{x}{s})\right]^2$: Squared Amplitude Product Device

CHART #11

Optical Realization of Electronic Amplitude Modulator

TRW



$$a(t) = \left[1 + m_a a_i(t)\right]$$

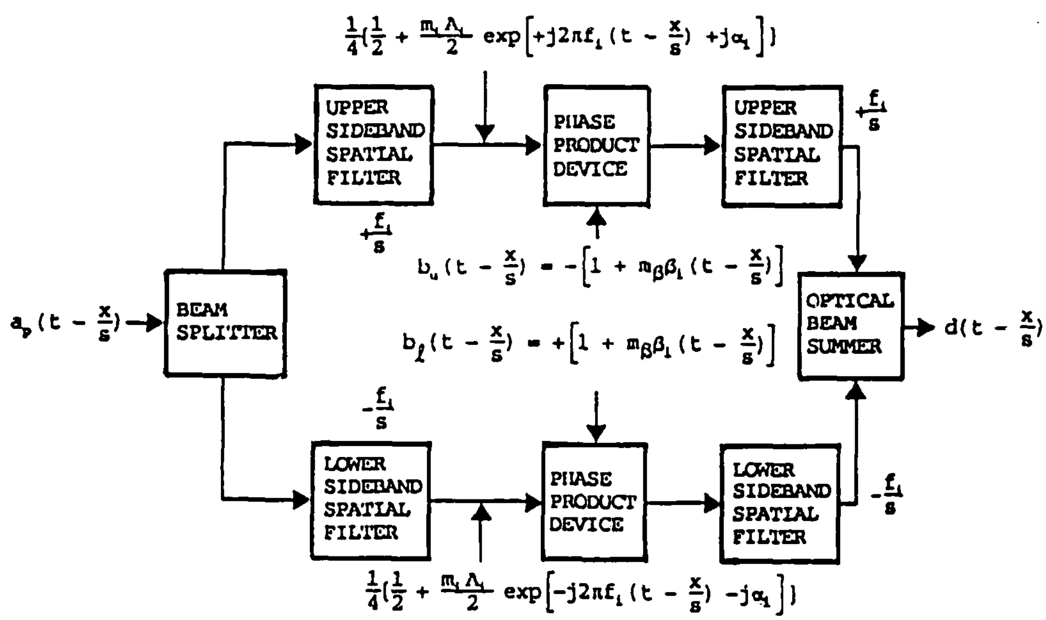
$$b(t) = \left[1 + m_i A_i \cos(2\pi f_i t + \alpha_i)\right]$$

$$v_o(t) = k_o\left[1 + m_a a_i(t - \tfrac{x_o}{s})\right]\cos\left[2\pi f_i(t - \tfrac{x_o}{s}) + \alpha_i\right]$$

CHART #12

Optical Realization of Phase Modulator

TRW



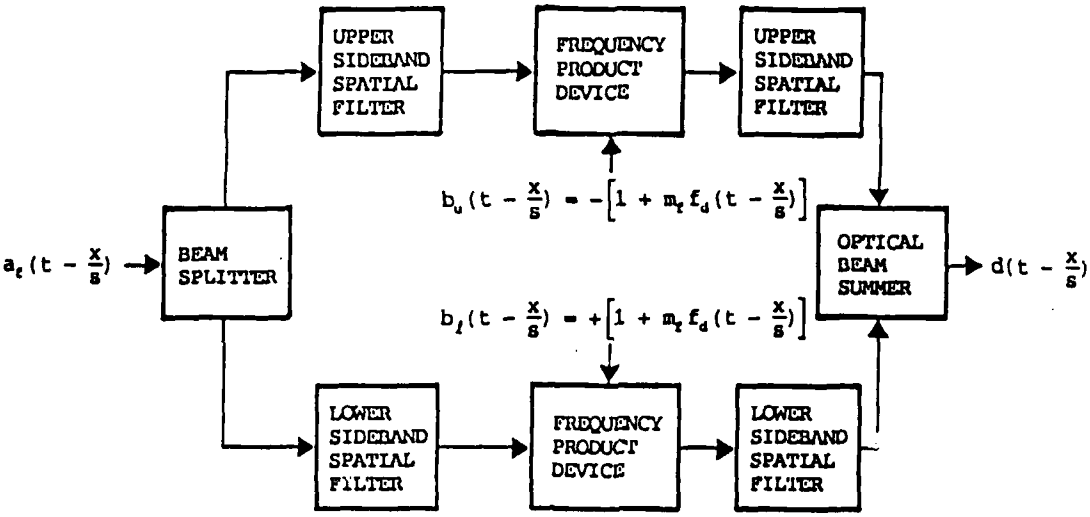
$$a_p(t - \tfrac{x}{s}) = \{1 + m_i A_i \cos\left[2\pi f_i(t - \tfrac{x}{s}) + \alpha_i\right]\}$$

$$d(t - \tfrac{x}{s}) = \{1 + m_i A_i \cos\left[2\pi f_i(t - \tfrac{x}{s}) + k_\beta \beta_i(t - \tfrac{x}{s}) + \alpha_i\right]\}$$

CHART #13

Optical Realization of Frequency Modulator (Voltage Controlled Oscillator)

TRW



Optical Realization Of Frequency Modulator (Voltage Controlled Oscillator)

$$d(t - \frac{x}{s}) = 1 + m_i \Lambda_i \cos\left[2\pi f_i (t - \frac{x}{s}) - \frac{2\pi}{s} \int_0^x f_d(t - \frac{x}{s}) d\gamma + \alpha_i\right]$$

$$a_\ell(t - \frac{x}{s}) = \{1 + m_i \Lambda_i \cos\left[2\pi f_i (t - \frac{x}{s}) + \alpha_i\right]\}$$

CHART #14

Electronic Implementation of Foster-Seeley Discriminator

TRW

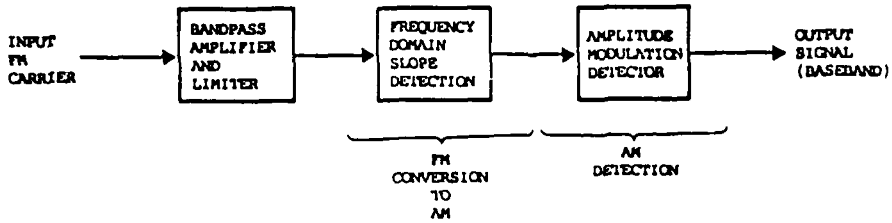


CHART #15

Magnitude of Fourier Transform of Optical Waveform Output of Acousto-Optic Modulator

TRW

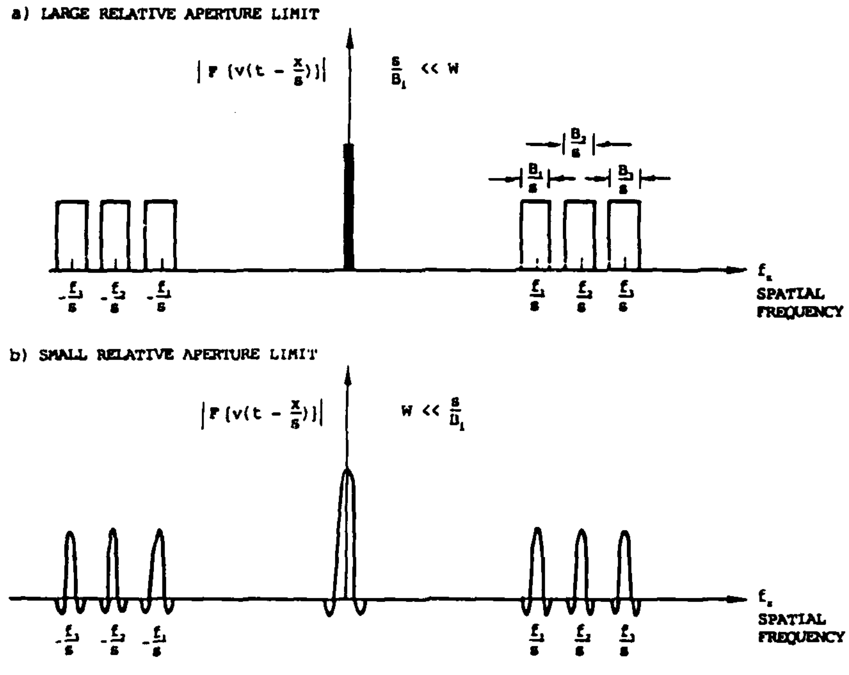


CHART #16

Optical Implementation of FM Discriminator

TRW

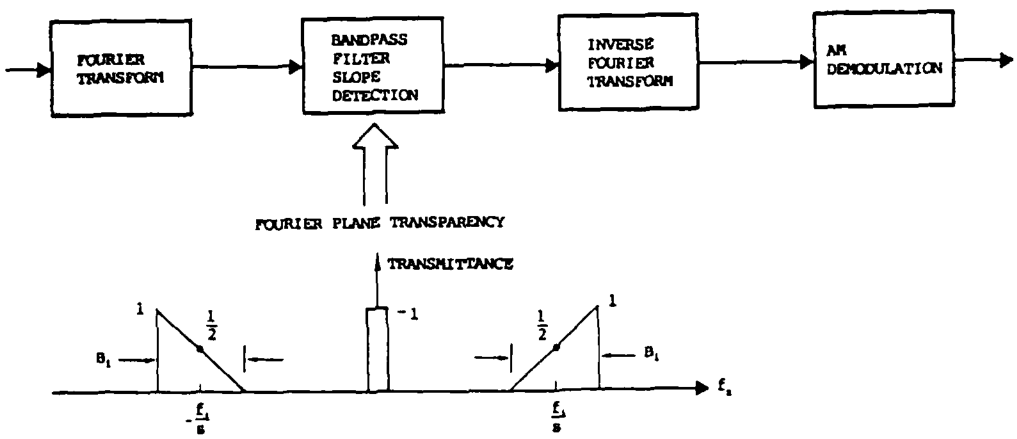


CHART #17

TRW

Model of All-Optical Phase Locked Loop

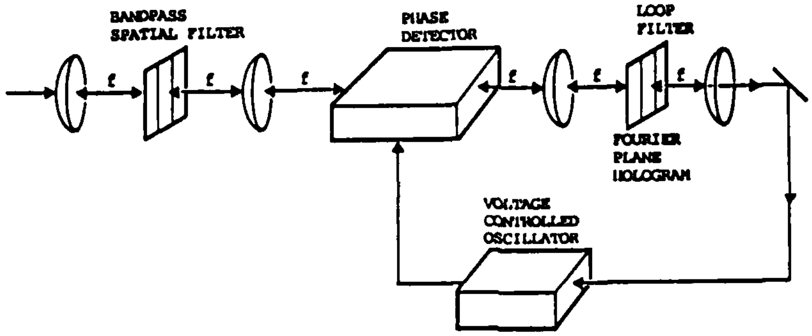


CHART #18

TRW

Phase Locked Loop Waveforms

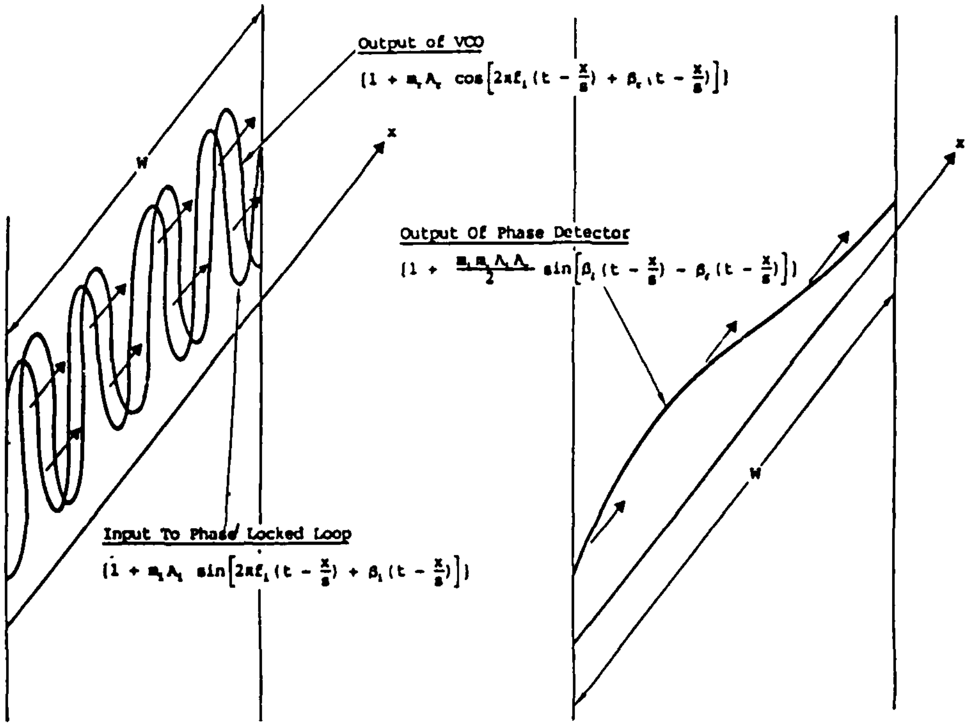
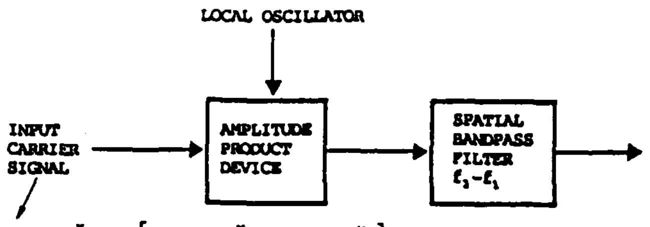


CHART #19

Optical RF Mixer

TRW

$$b_s(t - \frac{x}{s}) = 1 + m_2 A_2 a_2(t - \frac{x}{s}) \cos\left[2\pi f_2 (t - \frac{x}{s}) + \beta_2(t - \frac{x}{s})\right]$$



$$a_s(t - \frac{x}{s}) = 1 + m_1 A_1 a_1(t - \frac{x}{s}) \sin\left[2\pi f_1 (t - \frac{x}{s}) + \beta_1(t - \frac{x}{s})\right]$$

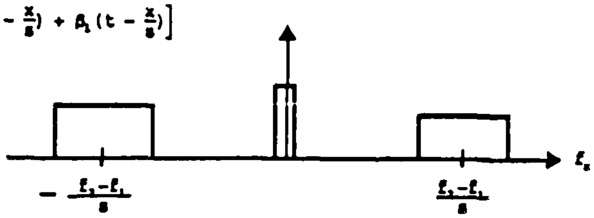


CHART #20

TRW

Devices For Acousto-optic Spatial Light Modulation

Produce Double Sideband Amplitude Spatial Modulation

1) Raman-Nath Cell With 90° Phase Shift Of Spatial Carrier
2) Dual Bragg Cells Combined For Upper And Lower Sidebands
3) Raman-Nath Cell For Spatial Phase Modulation Followed By Spatial FM Discriminator (And Integration).

Raster Scan Of Multiple Devices To Provide Longer Observation Time In Folded Format. Two Dimensional Device Of This Nature Represents Major Technology Breakthrough.

CHART #21

Raman Nath Mode Acousto-Optic Modulator for Spatial Double Sideband Amplitude Modulation

TRW

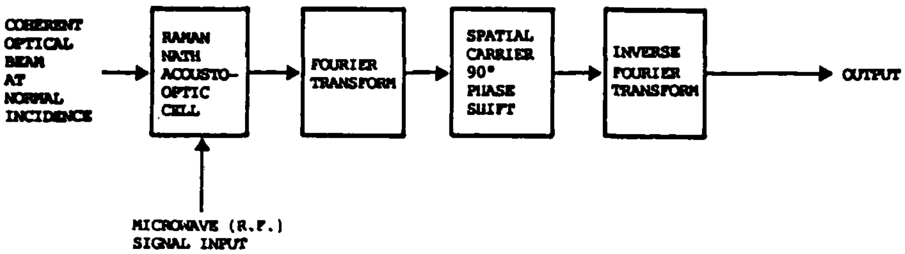


CHART #22

Bragg Cell Implementation of Spatial Double Sideband Amplitude Modulator

TRW

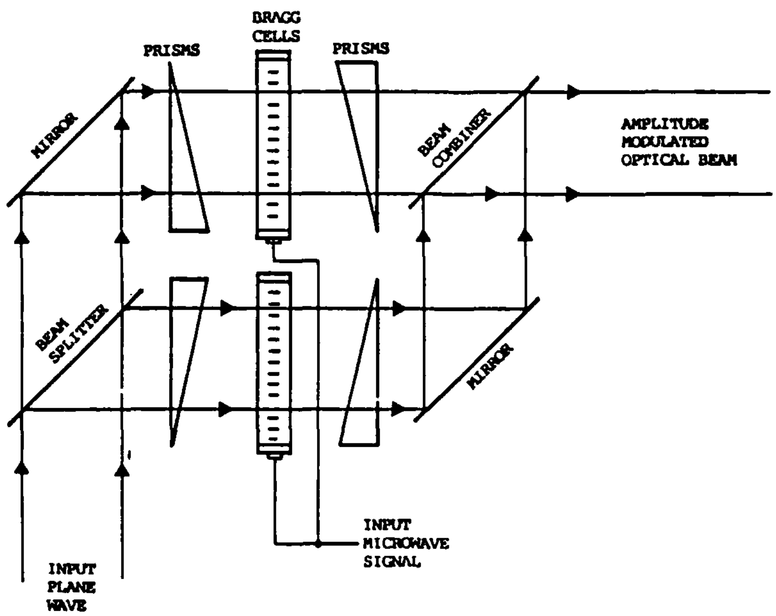


CHART #23

Mathematical Model of Bragg Cell

TRW

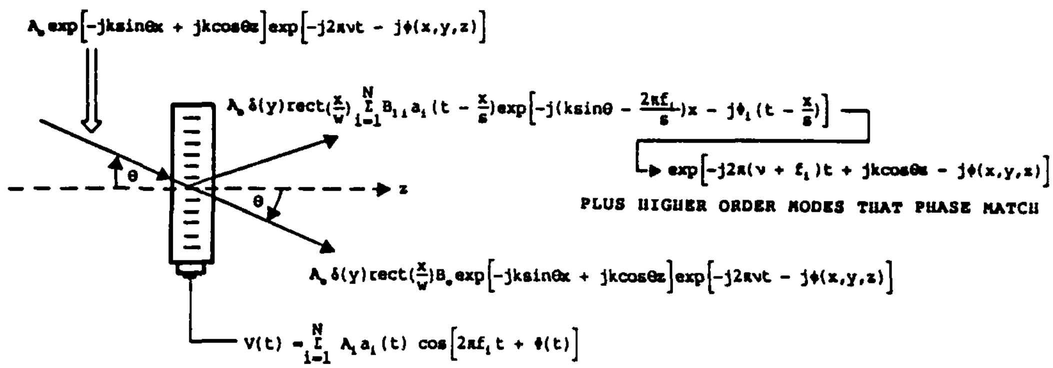


CHART #24

Linear Acousto-Optic Double Sideband Spatial Amplitude Modulator

TRW

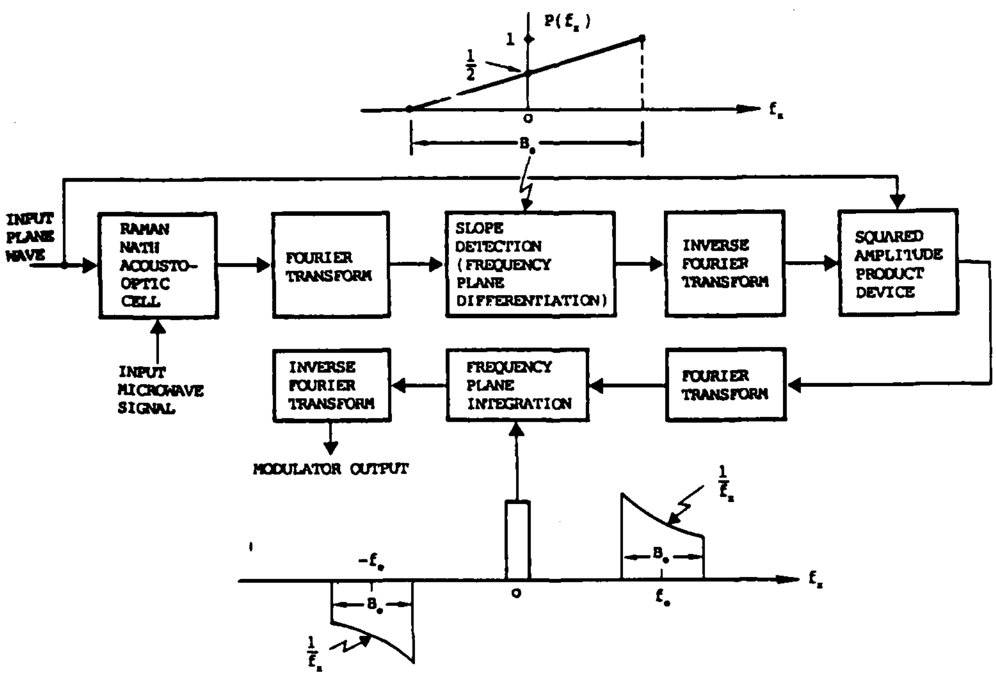


CHART #25

Raster of M+1 Acousto-Optic Cells

TRW

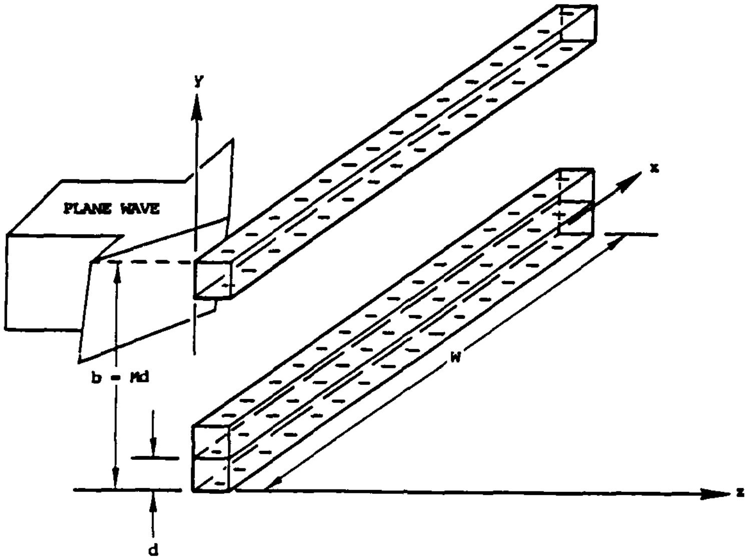


CHART #26

TRW

Optical To Optical Modulation Devices

- Photorefractive Device: Spatial Derivative of Refractive Index Proportional To Changes In Intensity of Modulating Beam (Becomes Spatial Voltage Controlled Oscillator).

- Analogous To Acousto-optic Devices In Refractive Index Variations. Can Provide All Four Hypothetical Devices.

- Phase Conjugator In Four Wave Mixing Configuration Provides Direct Beam Complex Amplitude Multiplications For Amplitude Product Device And Squared Amplitude Product Device.

CHART #27

Geometry of Degenerate Four Wave Mixing for Amplitude Product Device and Squared Amplitude Product Device

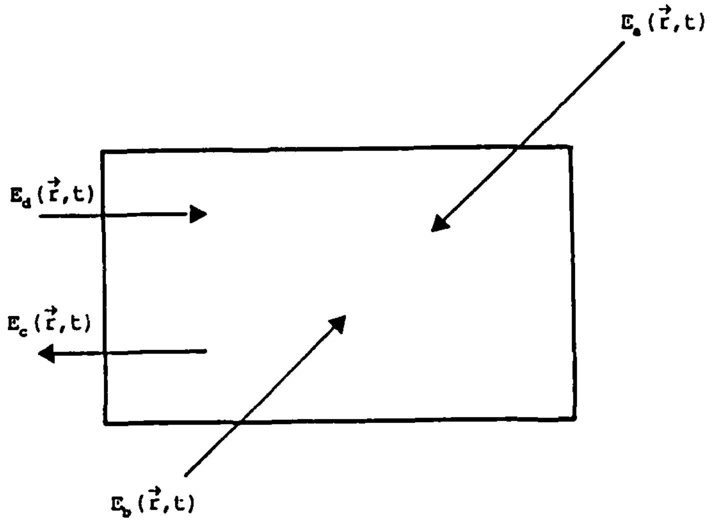


CHART #28

Phase Conjugation

$$E_d(\vec{r},t) = \mathrm{Re}\left[\Psi_d(\vec{r})e^{j(\omega t-kz)}\right]$$

$$E_a(\vec{r},t) = \mathrm{Re}\left[\Psi_a(\vec{r})e^{j(\omega t-\vec{k}\cdot\vec{r})}\right]$$

$$E_b(\vec{r},t) = \mathrm{Re}\left[\Psi_b(\vec{r})e^{j(\omega t+\vec{k}\cdot\vec{r})}\right]$$

$$E_c(\vec{r},t) = \mathrm{Re}\left[\Psi_d^*(\vec{r})\Psi_a(\vec{r})\Psi_b(\vec{r})e^{j(\omega t-kz)}\right]$$

CHART #29

TRW

FM Detection: Optical Implementation of Foster-Seeley Discriminator

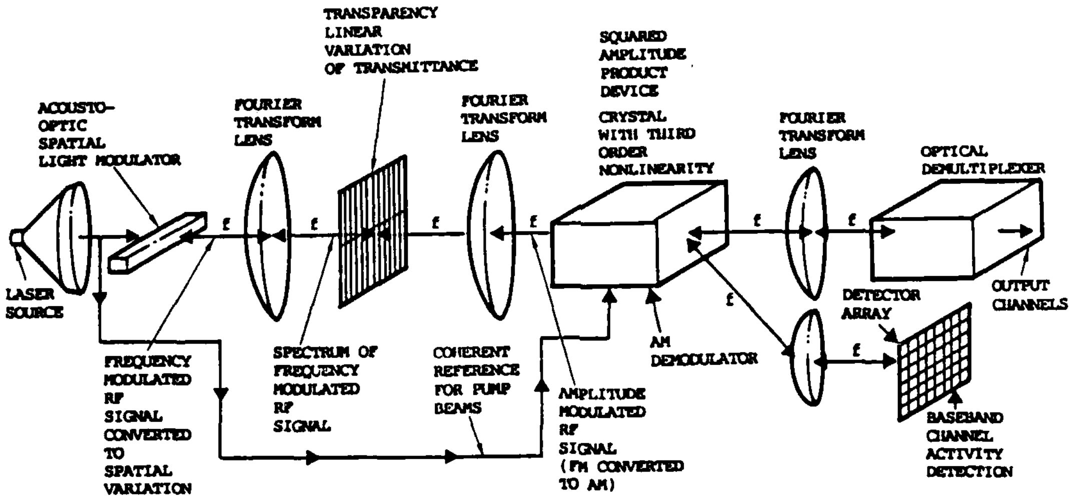


CHART #30

TRW

Optical Demultiplexer for Frequency-Division-Multiplexed Signal

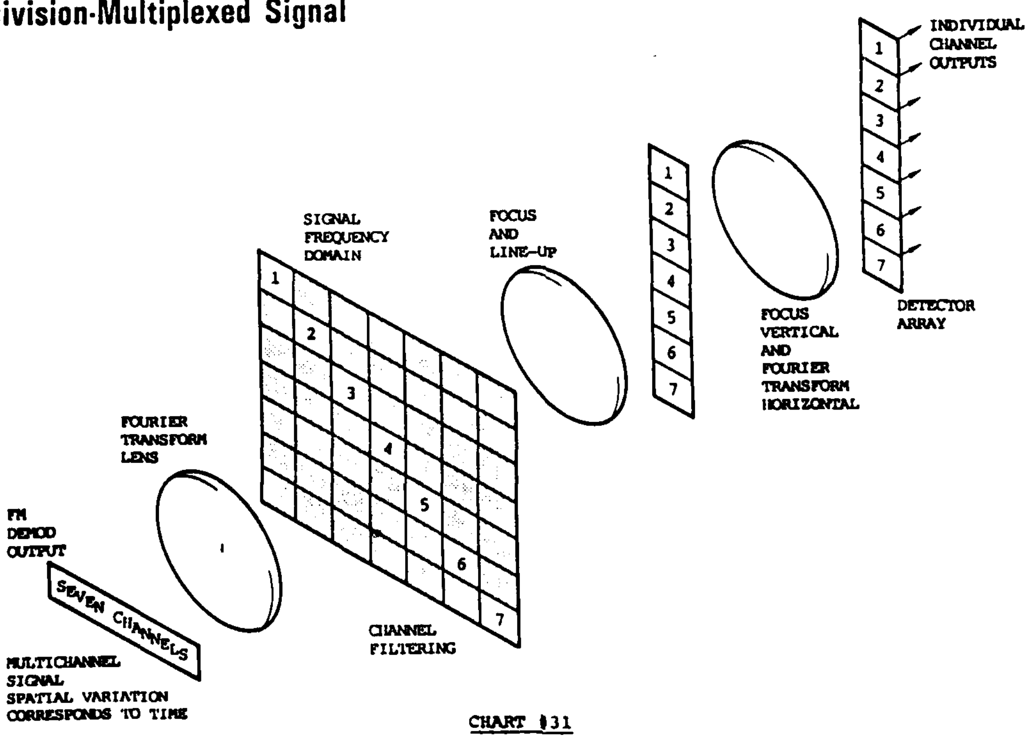


CHART #31

Optical Implementation of Electronic Phase Locked Loop

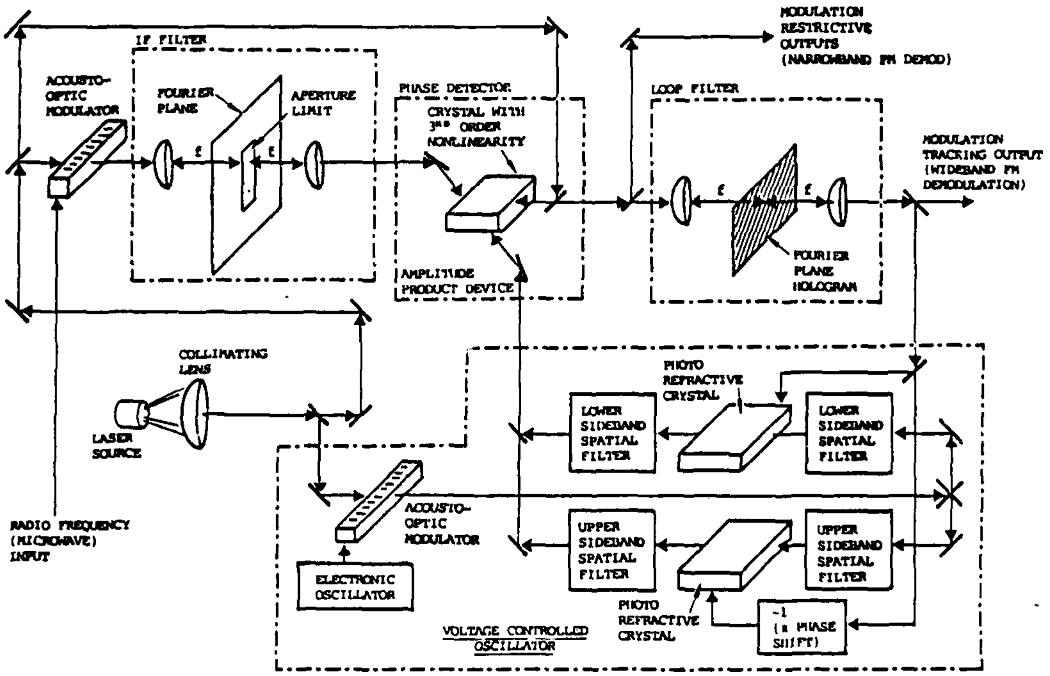


CHART #32

Optical Implementation of Optimal Processor

Optical Implementation Of Optimal Processor: Detection Of Pulse Signal With Unknown Phase, Amplitude, Frequency, Time Of Arrival, PRF, Pulsewidth

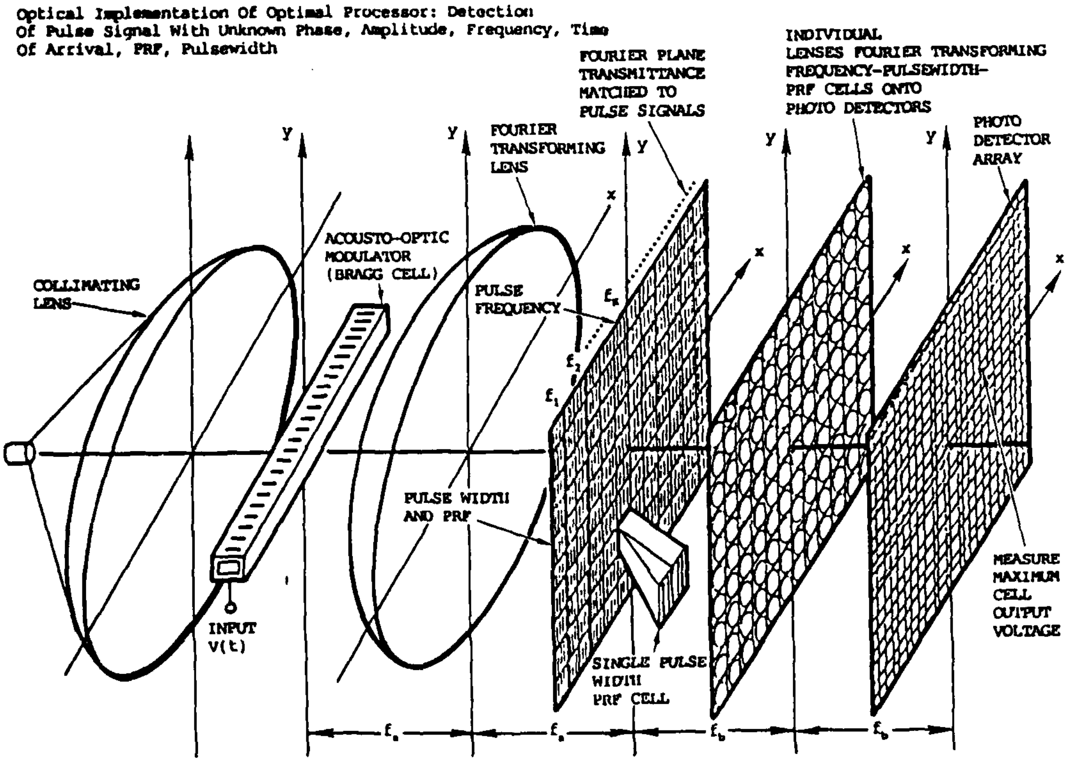


CHART #33

Matched Filter for Single Frequency/PRF/Pulsewidth Cell

TRW

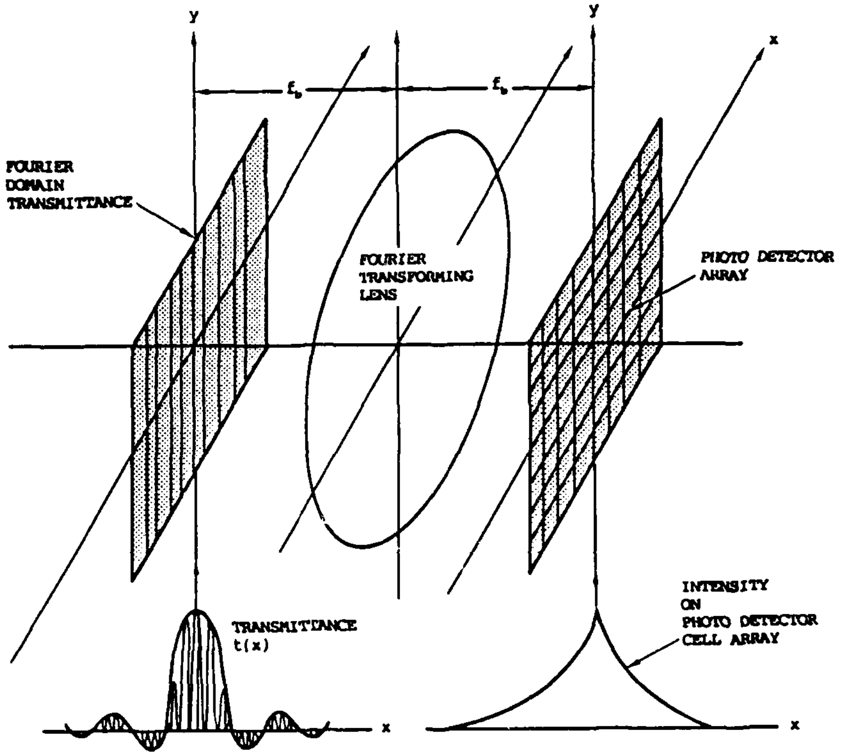


CHART #34

TRW

Main Contribution

- Architectures For Implementing Electronic Systems Utilizing Optical Processing Techniques And Components.
- Initial Effort Toward All-Optical Realization Of Complex Electronic Systems.
- Bridging Between Communication Theory And Modern Optics, Employing Linear And Nonlinear Techniques.
- Configured AM, PM and FM Modulators And Demodulators, Phase Locked Loop, RF Mixers, Filters And Matched Filters.
- Provides Some Guidance For Optical Device Research.
- Communications Satellite Multiple Carrier Demodulation/Remodulation Transponder Is Major System Result.
- Long Term Goal: Integrated Optics With Entire Electronic Systems Synthesized Within Bulk Crystals

CHART #35

TRW

Some System Advantages of Optical Processing

- Ease of performing Fourier transform
- Second spatial dimension for parallel processing can provide capability for exhaustive search of signal space
- Channelized communication system architecture matches optical two-dimensional switching and routing
- Can separately operate on positive and negative frequencies in Fourier transform domain
- Holography for amplitude and phase filter synthesis including generation from impulse response
- Inherent wide bandwidth, high carrier frequency, and fast response capability
- Rapid technological growth in applications and devices taking place

CHART #36

TRW

Major Processing Limitations

- Spatial Light Modulator Aperture Limit Becomes Microwave Signal Observation Time and Frequency Resolution Bound.
- High Data Rate Digital Signals Would Not Encounter These Limits Unless Time Demultiplexed
- Threshold Devices And Limiters For Decisions And Quantization Not Investigated.
- Typical Analog System Accuracy Limits Would Be Prevalent.
- Device Performance In Terms Of Bandwidth, Efficiency and Repeatability A Major Challenge.
- Materials For Integrated Optics With Properties: Acousto-optic, Photorefractive ‹ ıd Amplitude Nonlinearities.

CHART #37

Fourier plane hologram, which provides amplitude and phase functions as desired. The outputs appropriate to both modulation restrictive and modulation tracking operating modes of the loop are indicated. The voltage controlled oscillator utilizes the architecture of [CHART #14], with protorefractive crystals for the frequency product devices. The optical equivalent of a local oscillator is implemented by an electronic oscillator and the defined version of an acousto-optic modulator. This phase locked loop implementation is a major element which can be used extensively in the optical processing of microwave signals.

[CHART #33] shows the optical implementation of the optimal processor for the detection of pulsed microwave signal of unknown amplitude, phase, frequency, pulse width, pulse repetition frequency (PRF), and time of arrival. The electronic equivalent of this processor is depicted in [CHART #5]. In the optical version the matched filter functions are implemented in the Fourier plane, probably with a complex single hologram. The individual filter outputs are imaged on the photo detector array with a set of individual lenslets which constitute a fly's eye lens. An individual filter, lenset and array partition which forms a single matched filter is shown in more detail in [CHART #34].

[CHART #35] summarizes the main contribution of this research. It constitutes an architectural approach for implementing electronic systems with optical processors. It represents an initial effort toward all-optical realization of such complex systems. It is a necessary step, and hopefully an important step, in the evolution of integrated photonics wherein entire systems can be synthesized within bulk crystals.

The final two [CHARTS #36 and #37], summarize the major system advantages and limitations of optical processing for analog communication signals.

SASTRY: Understandably, this morning, we are concentrating on relay satellite applications, and so the concerns about electronic warheads and things like that. I have a suggestion for another application. In Rockwell we are interested in multiple satellite networks with laser links, and one problem we are looking at is the relaying of these messages at the inter-media satellites, and the relaying requirements are the source-destination requirements are embedded in the baseband signaling in the laser beam itself. So you need to get back to the electronics each time, in each satellite, before you could redirect to the appropriate route. But it seems to me now, we have a sufficient bag of tools by which, without requiring the electronic baseband signal, we should be able to do it optically, for example, use optical correlators, or use CDMA type things in which the ultimate destination should be conveyed, and perhaps it is some sort jiggling an assignment of the CDMA codes you might be able to have some sort of a routing selection process that could be done optically which would be of tremendous advantage. In that case, your electronic warheads would be much much faster. So we are looking at this optical signal processing at the intermediate satellite. Of course the destination satellites have to get back to the actual electronic signals. It certainly would be interesting to looking at some sort of system features of these aspects.

SULLIVAN: That is one of the questions that I thought about a fair amount. What are the input word-like light beams. Remember there is this thing where I could

come in as a light beam. Then you have to do what I am trying to do. I am using spatial dimension. You have to convert that. You have to get to something like that sound velocity. Something like that to be able to do that. And that is tough. But we would really like to be able to do that because it could use all kinds of laser links of all kinds around. Now what Bob was doing was his switching and routing. Clearly you could do that. But my type of thing is trying to do spatial processing and spatial domain. You have to get to it somehow and I couldn't figure out how to do that.

REIFFEN: I'll direct that to anyone of the panel. The thrust of what I understood all of you to be saying is that the way you have described optical signal processing is to convert time into space and then take the two-dimensional properties of optical stuff to do good things. That is a limitation of dealing then with only intensities and you have to work around that. I wonder if people who work this field have considered the fact that optical diodes can in fact be frequency modulated and demodulated by heterodyne systems, and I wonder if this kind of a technology has been thought of for optical signal processing application.

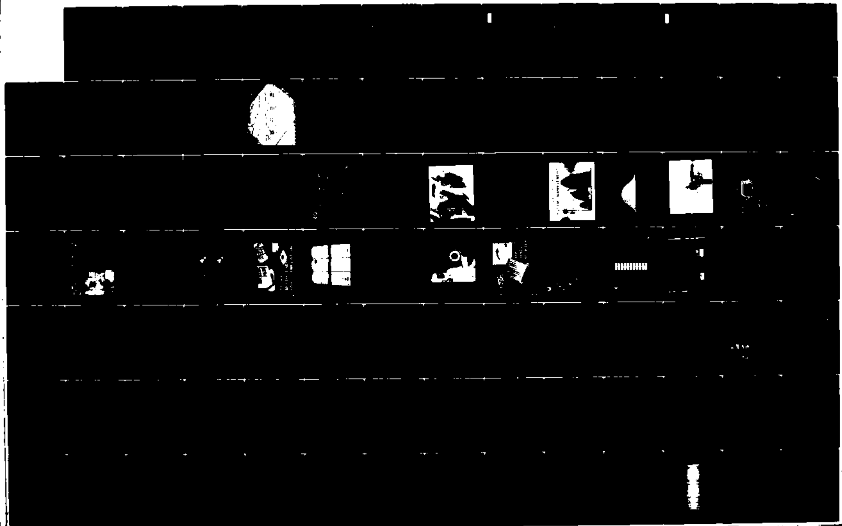
GAGLIARDI: You might be interested in reading one of the latest issues of lightwave technology that just came out about coherent optical processing of various types, both doing the parallel processing. It concentrates heavily on the parallel processing that was done basically done by the media set where you have coherent fields and heterodyne over a whole array of detectors simultaneously generate phases, simultaneous phases. You might be interested in reading that particular journal. It's the latest issue on lightwave technology.

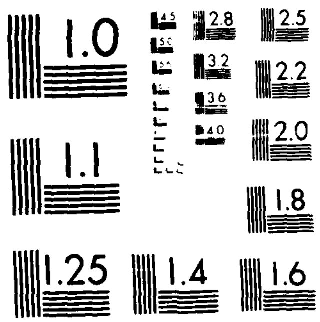
SCHOLTZ: That came out as a joint publication of the *Selected Areas in Communications of the Comm. Transactions*. So if you get the *Comm. Selected Areas Transactions*, you've got it.

HUTH: I was wondering, you were talking about resolution in the acoustic domain. And in the past when we have have looked at these kinds of things, basically you can spread the data. And so you don't have to worry about that, you might have some noncoherent combining maybe but you can get around the resolution problem just by spreading. So you don't have to worry about the megabit. The other thing is that I guess my concern in the past on signal processing is that I've seen a lot of it in laboratories, but I've seen little in actual systems. I think eventually we will get it to the point where you can use it exactly as you are talking about, being able to integrate it onto a substratum or somehow getting it onto a bulk type of integration. Until you can do that it's going to be in the laboratory state and it will state for a long time until you really get that in. I've seen so many experiments on these kinds of things where tremendous advantages have been shown in the laboratory but there is always this senior scientist who is tweaking, and yes, he can eventually make it all show up pretty nice. But when you are supplying something in a satellite system and things like that, to be able to do that, it has to go on to the next integration level.

DUPREE: I had an addition to the question that Barney had about heterodyne detector. The literature shows a number of different schemes that do use heterodyne detection. The simplest most straightforward way would be to set up the acousto-optic spectrum analyzer and a multiplexer

AD-A194 336 ADVANCED COMMUNICATION SYSTEM ENGINEERING PROCEEDINGS 4/5
WORKSHOP HELD IN SE.. (U) UNIVERSITY OF SOUTHERN
CALIFORNIA LOS ANGELES COMMUNICATION S.. R A SCHOLTZ
UNCLASSIFIED MAR 88 CSI-88-03-02 ARO-24913.1-EL-CF F/G 25/5 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

interferometer. So that you have an alternate path for plane wave reference source. The acousto-optic cell would operate at a higher frequency and in the Bragg cell mode or on an on-and-off mode with suppression of one sideband there would be a doppler shift coincident with the spectrum analysis. So that with an array of diode detectors, it would be possible to have the heterodyne source and tangent at the correct angle to get good detection efficiency of a variety of IF frequencies. The problem would be that it would tend to be temperature and vibration sensitive, since there is not an identical output path. Some other schemes and limited applications devise the optical path so that some of the undiffracted component of the beam from the passage through the A-O cell is ultimately redirected into the same direction and location angle as the diffracted component. Usually these involve combinations of two A-O cells, this is done in space-integrating and correlators. And in that case you use the undetected component as the heterodyne source. And so you find a number of those, but I think the problem that Dan has been dealing with here, is to try to deal with the general approach to the architecture. And in doing that, you sometimes have to give up some of the specialized applications. Ultimately you can hope to be able to do heterodying detection, but that remains to be worked out.

BOOTON: Dan, I have a question. Were you going to speculate as to how long it will be before you will see an actual flight payload, built in an optical form?

SULLIVAN: I don't think we are going to design it, Dick. Let me say, I think that one could apply something in 5 years. I think you can start something now and do some pieces of something and go ahead and apply it. I don't think you can replace all the electronics on that, but you could start. And I think there is a fair amount of interest on that. I think that for purposes of looking at many, many signals and things like that, and applications there, I think you can come up with a system, start now, and develop the hardware phase in a few years. You can do it.

HUTH: I think the things that Bob was talking about where you're just doing transferring, cross-switching and things like that, you could do that very soon. But when you start talking about the more esoteric things, that's a little more difficult. But I think the things that Bob is talking about are fairly close to being realized because you are working in mainly fibers and it is not as difficult as trying to integrate things together.

DUPREE: I thank you very much for your time and interest.

USC-CSI WORKSHOP ON ADVANCED COMMUNICATION SYSTEM ENGINEERING

Proceedings of Session Four: A Review of the State of the Art in Computer-Aided Design for Communications

CHOMA: I would like to welcome you to this afternoon's session. I am John Choma, and we are going to dispense with a lot of the formal introductions and so forth. I understand that there will be a photography session at 4:30. So we are going to try and wind things up by that time if at all possible. I've asked my speakers to adhere to a 20 minute discussion period, followed by about 10 minutes worth of questions, and hopefully that will allow us some time at the end for some interactive discussion. So I think without further ado, I would like to introduce to you Don Calhoun of Hughes Aircraft Company. Don has some teaching experience, has been at HAC for some 20 plus years, and he will be talking about some VLSI issues.

CALHOUN: I appreciate the opportunity to be here. This is a good group and the topic is certainly of strong interest to us at Hughes, and I am sure to Industry in general. I think this is also a good opportunity for USC to initiate efforts in this critical area and to bring academic attention to it. I thank Professor Scholtz for this opportunity.

I'd like to first give a little VLSI background. My background is very strongly on the digital side. We are trying to pick up very heavily in the analog, realizing the importance. A lot of the importance of it has become a situation where we've put so many things in place in terms of digital VLSI and VHSIC, that we are left with these ity-bity analog circuits that are absolutely critical, yet relatively risky to develop. What used to be just some real critical analog circuits now have a much higher proportion of risk and cost than their digital counterparts.

Let me show you just an example with this photo (FIGURE 1) of what we have done in digital. It is a chip of about 45,000 transistors which is about 3/10ths of an inch on an edge using 1.25 micron CMOS technology. It's an algebraic time-domain encoder/decoder, that had been about 4 circuit boards of electronics in a previous implementation with MSI circuits. We used AT&T as the supplier, and it has been tested at 40 MHz over mil temperature range. A key characteristic of this chip, which approaches the analog design problem, is that a single clock buffer fans out to something just over 1,000 flip-flop elements on that chip. Through analysis of the simulation, layout and back annotation, we were able to confirm that the maximum clock skew is 3/10ths of 1 nanosecond across that entire chip between any two flip-flops. To do that, you basically have to get into a strip line type approach of layout. I think that's quite a CAD achievement for what had been basically logic-level digital design previously. Consequently, thanks to some good suppliers that have really paid attention to computer-aided design, and thanks to the basic inherent characteristics of digital technology, we're able to put in place CAD systems now that can allow the design of circuits like this. That circuit is a standard cell implementation and several companies have put in place the computer-aided design tools that allow practitioners like ourselves to design these special cir-

cuits with reasonably low risk. In fact, in this case, the total design and fabrication was done in just under a year, and it was under a fixed price agreement to build the first circuits, and the first silicon was acceptable. We need those types of characteristics for analog.

In the analog area there are a lot of tools, but I would offer that they're basically independent task oriented tools. We don't really have the ability to say we've got an integrated set of tools, especially those that allow us to start with a requirement and end up with tested silicon and a test that have been implemented through the computer-aided design process. So there are some good tools, but they're not an integrated set. It's not a system of CAD tools that we can work with. There really are some reasons for that, though. Our comm-systems as indicated in FIGURE 2 really stress the analog to the extent we need highly specialized low noise and wide dynamic range with high precision and low tolerance limits that have to be achieved in the circuit, and we're operating up into the gigahertz range. So we're really pressing those areas. Where we've done reasonably well in terms of analog CAD is down in the lower frequency ranges where the tolerance limits and precision limits can necessarily tolerate larger spreads. I think the other observation is clearly that analog by its inherent nature doesn't have the ability to be looked at in such a discrete fashion as digital, with only a few variables that control the ability to simulate it. The characteristics that I see of the analog CAD today as summarized in FIGURE 3 is that the best CAD in the analog areas are really based upon regular components, structures or arrays of some kind. Such regular structures have not yet supported the analog requirements in our high performance communications equipment. Also we're dealing with arrays or component structures with discrete component values. Often we need a pretty continuous range of those values as we design the next application. Living with only a few of the discrete values for components is quite a restriction. And we seem to generally have had insufficient modeling in our design to allow the high performance to be met on the original silicon. We end up with layout sensitive designs and we often end up with fine tuning on the ultimate circuit when it's embedded in its next level of packaging.

Generally, as I mentioned, the tools are not integrated. It's not like we just haven't paid attention to it, it's really that we have an animal in the analog domain that's quite different from what we've dealt with in digital. With the digital we've principally worked with ASIC in the areas of gate arrays or standard cells, or things like structured arrays or structured cells, where we take different levels of functional regularity, at the gate level, transistor level or some block level, and integrate that across a chip. We have a hierarchy from which we can build blocks as large as entire microprocessors which then become part of a chip. That building block structure in the digital world doesn't have a counterpart yet in the analog domain. We lack that ability to have a hierarchy, and it's really key.

Probably the wish list that we'd like to create is one with parameterized blocks where the floorplan of the chip would be arbitrary. In other words, we could take out of a library the blocks that would be required in an application. We could actually give them a value set based upon the CAD supported design that would cover a full continuous range for the requirements. I would think that this would be better than the standard blocks and better than the

other approaches. It still has restrictions in meeting the very high performance and the very tolerance-sensitive requirements, but I would recommend attention in this area, especially as supported by behavioral simulation. I think that's an area where we really have to progress more in the analog world. We need to present to the designer at a workstation frequency dependent behavioral analysis capability and let him proceed in his basic design so we can get early prototypes at reduced design risk. We still may be at a point where we've got a lot of fine tuning and design modifications, even after that first design. Analog CAD isn't more available largely because analog has so many interacting constraints which determine the performance of the circuit. An example here is that a SPICE model of a bipolar transistor has about 42 parameters to be specified. That's a huge number compared to what we do in terms of digital gate modeling. For instance, when AT&T goes from one process to the next generation process (which may happen every 18 months to two years), they can in a couple days on a VAX 11/785 regenerate the parameters of their digital library. So they have a library which can be software tuned to a new process and be verified against that. There really isn't a good definition of a hierarchy for the analog domain that supports the kind of design tools that we think of in the digital world. If we look at the sub-blocks in the digital world used to build the ASIC arrays and apply those to the analog domain, we really have a very large library of intended parametric or parameter specific reusable sub-blocks. The list of what all our applications would need would be very large in terms of the functional identity of those blocks and in terms of the various parameters each of us would choose from one application to another. As I mentioned, good analog designs generally require fine tuning, and testing is very application dependent. These two areas together cause us a horrendous amount of design attention, schedule, cost and risk in the development. At this time, we really can't bring enough attention to get the design right in simulation alone. We still need the allowance for fine tuning as the parts are produced. The testing itself in the analog domain doesn't have the ability like digital to do the fault analysis-fault coverage analysis similar to the digital model.

As I mentioned, we need the tool integration of the analog tasks; we also need the analog tools integrated with digital tools. We're coming to the opportunity where a lot of analog and digital can be combined effectively on chip. One thing that I see is missing, or at least is a few years slower in developing, is that we really don't have the degree of supplier support on ASIC products in the analog domain. Digital ASIC has grown into a very nice business with separate business units within many of the semiconductor companies, but not so much in analog. We especially don't see the analog suppliers as far along in terms of the ability to provide CAD tools to the customer. At the system design level of our communications hardware we really need support at the behavioral analog level for simulation and for mixed mode analog and digital simulation. We need analog to be part of the ability to simulate hardware at the higher levels.

So the conclusions, then, are that digital ASIC has become well supported in our communications hardware developments. Digital designs are well supported by computer-aided design tools and by suppliers in the industry that are really making a business of being there for supporting those products, taking fixed-priced contracts

HUGHES

COMMUNICATIONS HARDWARE

CAD STATUS AND REQUIREMENTS

DON F. CALHOUN
28 MAY 1987

Figure 1

GROUND SYSTEMS GROUP/FULLERTON, CALIFORNIA

HUGHES

COMM. SYSTEMS HARDWARE/ANALOG CAD INCOMPATIBILITIES

COMM. SYSTEMS FEATURES	CURRENT ANALOG CAD CHARACTERISTICS
• HIGHLY SPECIALIZED LOW NOISE AND WIDE DYNAMIC RANGE REQUIREMENTS	• BEST CAD BASED ON REGULAR COMPONENTS/ STRUCTURES
• HIGH PRECISION AND LOW TOLERANCE LIMITS	• DISCRETE COMPONENT VALUES OFTEN WITH PREPLACED COMPONENTS
• OPERATING FREQUENCY UP TO GIGAHERTZ RANGE	• INSUFFICIENT MODELING FOR HIGH PERFORMANCE AND LAYOUT SENSITIVE DESIGNS
	• GENERALLY TOOLS ARE NOT INTEGRATED

Figure 2

GROUND SYSTEMS GROUP/FULLERTON, CALIFORNIA

PROBLEMS IN ANALOG CAD

HUGHES

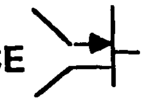
- MANY INTERACTING CONSTRAINTS WHICH DETERMINE PERFORMANCE (SPICE  = F (42)).
- NO DEFINED CONCEPT OF HIERARCHY.
- VERY LARGE LIBRARY OF PARAMETER-SPECIFIC REUSABLE SUB-BLOCKS.
- GOOD DESIGNS REQUIRE FINE TUNING.
- TESTING IS VERY APPLICATION-DEPENDENT.
- NEED TOOL INTEGRATION (INCLUDING A&D).
- NEED MAJOR SUPPLIER PRODUCT SUPPORT.
- NEED COMMUNICATION SYSTEM DESIGN SUPPORT WITH BEHAVIORAL ANALOG AND MIXED MODE ANALOG-DIGITAL SIMULATORS.

Figure 3

HUGHES

SEMICUSTOM ANALOG ALTERNATIVES

METHODOLOGY	EXAMPLES	FLOORPLAN	COMPONENT VALUES	COMM. SYS. HARDWARE
SINGLE COMPONENT ARRAYS	TRANSISTORS, RESISTORS, DIODES	PREPLACED	DISCRETE	MOST FLEXIBLE BUT LIMITS PERFORMANCE. SLOWS TURNAROUND.
*PARAMETERIZED COMPONENTS	TRANSISTORS, RESISTORS, DIODES	ARBITRARY	CONTINUOUS	NOT AVAILABLE.
CIRCUIT ARRAYS	DIGITAL GATES. CAPACITORS, OP. AMPS., SWITCH CAPACITOR BLOCKS	PREPLACED	DISCRETE	LESS APPLICATION. AREA INEFFICIENT.
STANDARD BLOCKS	OPS AMPS, COMPARATORS, D/AS, A/DS, OSCILLATORS	ARBITRARY	DISCRETE	HAVE TO DESIGN SYSTEM AROUND BLOCKS.
PARAMETERIZED BLOCKS	SAME AS STANDARD BLOCK	ARBITRARY	CONTINUOUS	BETTER THAN STANDARD BLOCKS. STILL RESTRICTED.

* REQUIRES DEVELOPMENT OF EXPERT SYSTEM.

Figure 4

HUGHES

WEAKNESSES IN ANALOG CAD FOR COMM. SYSTEMS HARDWARE

- INTEGRATION OF TASK ORIENTED CAD TOOLS
  - INTERFACE STANDARDS NEEDED
- BEHAVIORAL ANALOG SIMULATORS
- MIXED ANALOG/DIGITAL SIMULATORS
  - ADVANCED MIXED MODE (CIRCUIT, FUNCTION AND BEHAVIOR) ANALYSIS
- TESTABILITY OF HARDWARE
  - NEED TOOLS TO SUPPORT TEST PROGRAM DEVELOPMENT
  - ANALOG/DIGITAL?

Figure 5

GROUND SYSTEMS GROUP/FULLERTON, CALIFORNIA

CONCLUSIONS:

HUGHES

- DIGITAL ASIC HAS BECOME WELL SUPPORTED BY CAD AND SUPPLIERS DURING THE 1980'S.
- THROUGH IMPROVED PERFORMANCE, DENSITY AND RISK, DIGITAL IS SUPPLANTING SOME ANALOG.
- 5000:1 GAIN IN DIGITAL ASIC "FOM" SINCE 1982 (FOM = GATES/CHIP•SPEED•GATES/$NRE•GATES/PROD.$)
- ANALOG IS STILL REQUIRED, ESPECIALLY HIGH PERFORMANCE COMMUNICATIONS.
- WITHOUT INTEGRATED CAD AND MAJOR SUPPLIER SUPPORT, ANALOG WILL REQUIRE INCREASING BURDEN IN DESIGN EFFORT RELATIVE TO DIGITAL.

Figure 6

to source those products at low risk. As stated in FIGURE 4, through the improved performance, density and risk, digital has been developed to a point where digital really is replacing a lot of the analog activities that we've had in the past. We feel though that the analog is clearly still required, and it's starting to take more of a significant share of our total design effort because of how fast the digital technology is moving in this area. As an example, I see about a 5,000 to 1 gain in the digital ASIC figure of merit since 1982. I define that figure of merit as a product of "gates per chip" times "speed" times "gates that can be developed per dollar of nonrecurring cost" times "number of gates that can be procured per production dollar", and those four together have all gotten better in the last 5 years with a total product gain of about 5,000 to 1. You can probably find examples that are even higher than that. The analog then becomes more imperative because of the critical requirement we can't get around in high performance communications. We haven't seen that gain in the analog. Without the integrated CAD and the major supplier support, which I think are two key issues here, I see the analog requiring an increasing burden in our total hardware design effort.

SCHOLTZ: Is there a need or desire -- you mentioned CAD tool support coming from the suppliers to you -- is there a need in this area for say CAD for surface acoustic wave devices, for example? That's sort of out of the realm of the things you've been talking about I think.

CALHOUN: I clearly think there is, and I make the analogy that we've had digital ASICs since the late '60's. Fairchild, for instance, had a micromosaic array and such in the 67-68 time period. What happened in the twelve to fifteen years from there until the early '80's was that a lot of the CAD for digital was customer CAD, or third party CAD. The real difference that occurred the success of digital ASICs was when the customer could start using CAD that was well supported by a supplier, with it linked together so that there was minimum risk between the designer's understanding of that vendor's process and how to model his circuits. I think in surface acoustics we have the same thing. If I'm going to try to understand the characteristics of those devices and the supplier hasn't understood it, I have a more risky relationship with my supplier. I can think of a couple of companies which I think have really characterized their process. I can also think of some others where it's in the customers' hands to understand that vendor's process and characteristics thereof.

MOUFTAH: Do you have any CAD tools for digital design for testability?

CALHOUN: Yes, our company does. There are some pretty recent things that we've been doing internally which are being tested right now. I know of some of the test results that show some good promise in that area, and there are quite a few gradations in terms of performance. I don't think there's anything that's quite in the public domain though that I can talk to you about yet.

MOUFTAH: For example, do you have any sort of tools to apply LSSD techniques or scan path?

CALHOUN: Yes, that's incorporated in the tools that are completing development, and I think other companies have such as well. I know IBM uses tools, and I believe AT&T uses such tools. The problem with tools like that is that they're almost exclusively in-house developed and

proprietary.

MOUFTAH: IBM is doing a lot of work in this area. Also Bell Northern Research in Ottawa, Canada, is doing some work in this area.

CALHOUN: Overall, there is a lot going on inside companies, and some of this has been made public in terms of papers, but not in terms of programs that can be used by others outside.

REITMEYER: I could make a comment on that and I'll speak to that a little bit in terms of some of the things that VHSIC is doing during my talk.

CALHOUN: Question over here?

SHEU: Yes, I would like to know what you can foresee as the major impact of expert system on analog circuit design?

CALHOUN: I think there's a real opportunity there. I'm honestly new enough to it that I don't feel technically strong here, but a lot of the analog designs that we're doing should be able to benefit from the expert system approach. I think it's an area where we'd love to capture more of that knowledge and the procedures by which we can apply good design techniques through expert systems. I really look forward to solutions in that area. I don't think I can quantify an answer for you, but I really think that's a proper approach to take especially if it can be integrated with workstation-based approaches where the designer can be given some basic tools by which he can do the behavioral modeling, and then with the buildup of the expert system knowledge base, develop the rules by which the design can be more correctly done the first time. Any other questions?

CHOMA: Don, I have a question. You commented on the SPICE modeling of bipolar transistors; I assume that was developed in the analog model you're talking about, the 42 parameters. Clearly if you're designing something that has say 100 transistors on a chip, that becomes rather impractical, even 50 transistors I suppose is impractical. Do you see the modeling venture going more toward a kind of a standard cell-canonic cell type approach, where for example we would take a common cell, like a common-emitter, common-base cascade, characterize that as a device, model that in terms of a finite set of parameters and go on with block models?

CALHOUN: I really think it has to take that approach. I guess I'm reminded by yesterday's comment of a designer's capability to handle 10 elements a day. I mean we really can't stay with the device being at the transistor level and design large analog circuits. We've got to have the blocks, and we must build some blocks that are reusable in our designs.

CHOMA: Our next speaker for the day is Prof. Gupta of the University of Colorado, who's going to speak on microwave and millimeter integrated circuit CAD approaches. Prof. Gupta ....

GUPTA: Well, my comments this afternoon are restricted to the current state-of-the-art in microwave and millimeter wave CAD. By microwave CAD, I mean computer aided design of integrated circuits and antennas at microwave and millimeter wave frequencies. The status of CAD techniques in this area is quite different from that of VLSI design and of digital circuit design. It is even different from the analog circuit design at lower frequencies. The main reason for this state of affairs is that until recently, until a few years ago, most of microwave, millimeter wave circuits were hybrid integrated circuits, not monol-

ithic circuits. Recently, because of the progress and enhanced interest in the monolithic circuit design at microwave-millimeter wave frequencies and in monolithic integration of antenna elements in these circuits, techniques have been developed for a new generation of CAD tools which could carry out monolithic design at microwave and millimeter wave frequencies. I'll comment on some of these topics.

One of the problems (one of the bottlenecks) in successful designs at these frequencies is in the modeling of various lumped devices, both active and passive. First of all a picture of what I'm talking about. This is a monolithic version of a microstrip antenna system. This VIEWGRAPH (#2)[1] is not a very good one, it's borrowed from my friends at Ball Aerospace. It shows, basically, a 4x4 array. This is a unit cell of the array. The most prominent feature here is the microstrip radiating element. This is an amplifier, this is a 3-bit shift register, and we have the millimeter wave power distribution network of microstrip lines, feeding power to these elements of the array. In addition to RF-signal, we have various other bias and control lines. So far we do not have adequate CAD tools which will allow us to design this type of system. Several groups are trying the design of such systems but without good results yet. Similar systems using hybrid technology approach have been successfully designed. One of the recognized reasons is that we don't have the capability of tuning in these monolithic systems similar to what we have in hybrid systems. Because of that we need more accurate computer aided design tools. On the VIEWGRAPH (#3) here we show the Gallium arsenide chip which is a monolithic 2-8 GHz amplifier. It is the one unit here. This is the amplifier, this is the MESFET with the active bias, this is another MESFET which gives bias. The amplifier has a feedback path, the faint line which you can hardly see. Again, the design of these types of circuits requires very accurate CAD tools. I'll also comment on some necessary CAD tools that we lack today. But before I get to that, I usually show this VIEWGRAPH (#4) -- it's one of my favorite viewgraphs -- which outlines "What a CAD process is?" I'm sure most of you don't need such an introduction, but essentially I'll show it to point out the areas which need further work.

In any computer aided design process you start with a set of specifications, you have some synthesis method, some design data (many times computerized), to develop an initial design. You model that initial design to carry out a computer aided analysis. You compare the results of this analysis with the given specifications and quite often we see that the specifications are not met. So you go through this iterative loop of modifying the initial design, carrying out analysis again, and comparing with the given specifications. This is the computer aided optimization loop. In a typical circuit design one goes through this computational optimization loop something like 100-200 times, many times through the sensitivity analysis step in order to accelerate the optimization process. Of course, we use the sensitivity analysis also for calculating tolerances of a design. After specifications have been met or after we give up (saying that the specifications cannot be met), you pass it on to fabrication. Measurements are carried on the fabricated circuit and again you compare the performance with the given specifications. If our

[1] Viewgraph #2 is missing.

design had been successful, there would not be any need to modify it again at this final stage in order to meet specifications. But the way things are at the present state-of-the-art, at least in microwave and millimeter wave frequencies, it becomes necessary to go through these experimental iterations. Of course the whole purpose of computer aided design is to minimize or to eliminate these experimental iterations. These experimental iterations are costly in terms of both money and time; my friends in monolithic microwave industry tell me that it could easily take something like 6 months and about $100,000 to carry out one experimental modification. I could sum up this by saying that there are three aspects of any CAD process -- it works also in any other CAD -- modeling, analysis and optimization. I'll comment on how these apply to monolithic circuits at microwave frequencies ranges and to microstrip antennas.

The weakest link in the CAD process is in the modeling: modeling of passive devices, modeling of transmission structures, discontinuities in transmission structures and modeling of active devices. The analysis and optimization are more advanced aspects of CAD. As I just mentioned, the weakest link in microwave-millimeter wave CAD is in the modeling. At microwave frequencies, a microstrip transmission structure is most commonly used. The microstrip line at microwave frequencies is at a fairly good status in terms of its characterization in terms of characteristic impedance, phase velocity, attenuation constant, and dispersion characteristics. But these features are not known accurately at millimeter wave frequencies. More accurate models need to be developed and incorporated in CAD systems. Even worse is the situation when one looks at the discontinuities and junctions in the transmission lines. At micro-millimeter wave frequencies microstrip bends, T-junctions, changes in the impedance, etc. cause parasitic inductancies which need to be characterized accurately, otherwise circuit performance becomes very different from what we are looking for.

Another area where we're still looking for improved models is the nonlinear models for active devices, most commonly used device is Gallium arsenide MESFET. This VIEWGRAPH (#5) shows various types of discontinuities that occur in various circuits that are fabricated. We still don't have the accurate characterization of these structures. I'll go in a little more detail about that. If we are trying to characterize these specific structures, these discontinuities there are three different group of methods available to us. We have the electrostatic analysis which is essentially the calculation of extra capacitance or inductance associated with these discontinuities. Most of the present day design tools, CAD tools, use the results based on this approach. These results are put in closed-form expressions which are used in design. These expressions work quite well up to 12 GHz or 18 GHz, but if you go in higher frequencies their accuracy decreases. (VIEWGRAPH #6) At the second level we have what we could call planar model analysis, which uses a planar model of the microstrip line and carries out a characteristic S-matrix analysis of these junctions. More accurate fullwave analysis methods are also available. The most commonly used is Spectral Domain Analysis. These are numerical methods which give you results for a specific set of parameters. However, very few results have been reported so far, and so as far as I know, none of these has been integrated with available CAD systems today. I'll skip

some details. Let me give you some more details about this planar model analysis. I'll just briefly point out what this method is all about.

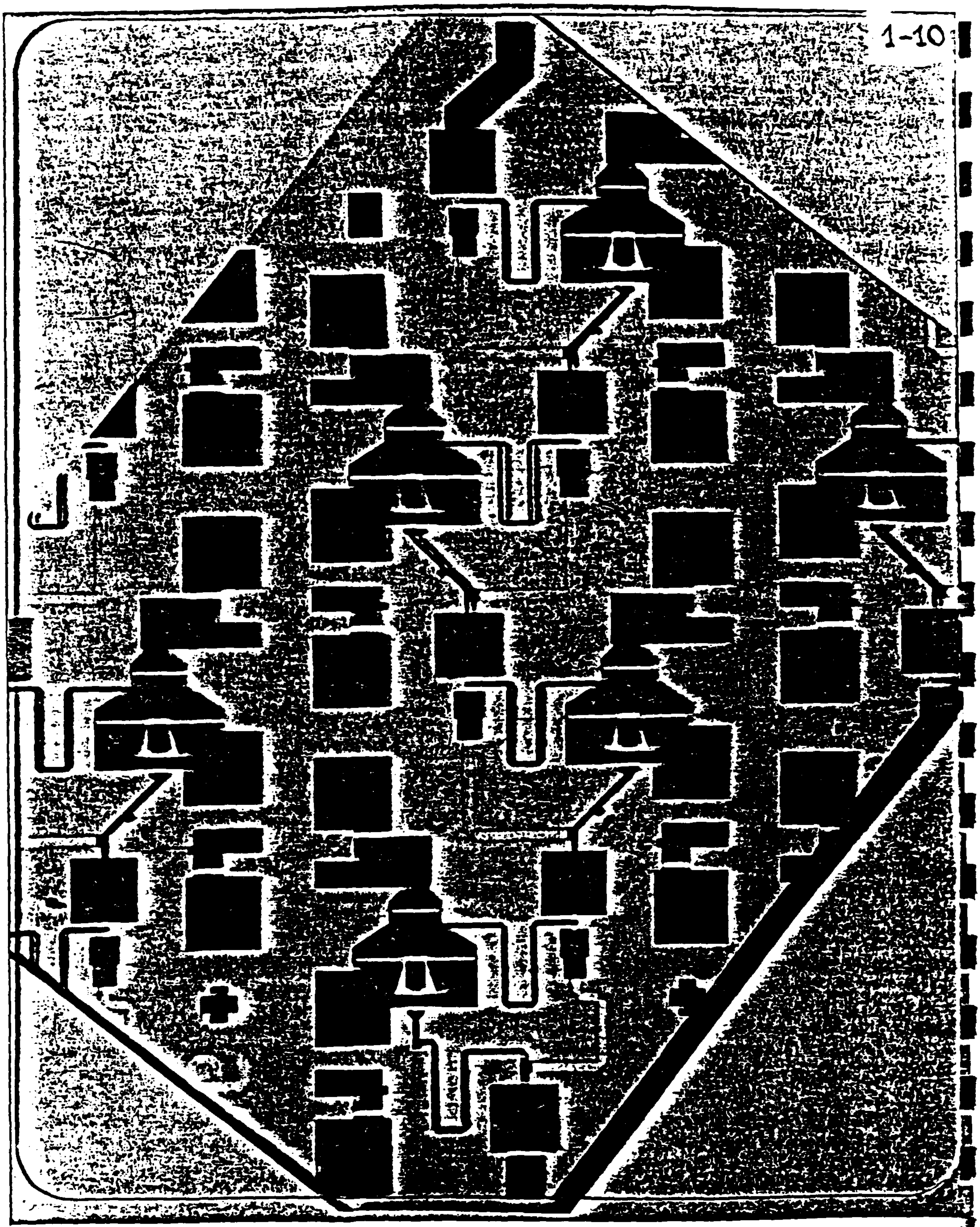
(VIEWGRAPH #7) Suppose we're having a junction of three lines of different lengths and different impedances. You want to characterize this junction in terms of scattering parameters (which are functions of the frequency) over the range of frequencies we are interested in. This planar model approach transforms it into a planar waveguide model that means the fringing field at these edges is accounted for by having an effective width for these lines, and an effective dielectric constant of these lines. One could consider this as a planar 2-dimensional structure and carry out a 2-dimensional analysis. In this case, it's shown by considering the finite lengths of these three lines and by dividing it into three rectangular segments. Each one of these rectangular segments can be characterized in terms of a multiport characterization and then you use multiport network techniques in order to put them together into arriving at an overall scattering matrix characterization. There are some computer programs available for carrying out this type of analysis, but these computer programs are not fully integrated with simulation and analysis programs. One of the difficulties in this integration is that the results we get from this detailed electromagnetic analysis are numerical in nature. And there are difficulties when you want to include those numerical results or numerical models in computer aided design process. What is being illustrated (VIEWGRAPH #8) is that one can use something like expert system or expert database type of concept in combining simulation programs and rigorous analysis programs for discontinuity analysis in the CAD process. What one really needs is a multidimensional interpolation from data files that have been created by this discontinuity analysis program. At some stage during the analysis process, if the set of parameters is such that the data is not available in data files, this program should be able to update data files by calling this more time consuming program and storing that information and interpolating from this information. I guess that's one of the areas where we want to make use of some concepts of expert systems for CAD as indicated here. Another associated problem is that not only do we need to characterize this type of discontinuity at junctions but it's also desirable, in order to improve the circuit's performance, to modify the configurations of these discontinuities so that the parasitic reactances can be minimized. In other words one could, by modifying the geometry of this junction, arrive at an optimum configuration which has the least parasitic reactances. There are some limited results available here again, but there is need again to integrate these things in the CAD procedure.

One of the current trends in microwave/millimeter-wave CAD is the use of vector processors (or supercomputers) which are becoming more and more accessible these days. Because of the repetitive nature of the computational steps in the various algorithms for microwave CAD, use of vector processors results in computational efficiency and reduces the computational cost. One of such cost comparisons is shown in this VIEWGRAPH (#10). This is based on a recent paper in *IEEE Transactions on Microwave Theory and Techniques*. The horizontal axis on this plot is the number of frequency points where the microwave circuit is being analyzed. It is seen that the use of supercomputers is more

TRENDS IN MICROWAVE CAD

K. C. Gupta
Electrical & Computer Engineering Department
University of Colorado
Boulder CO

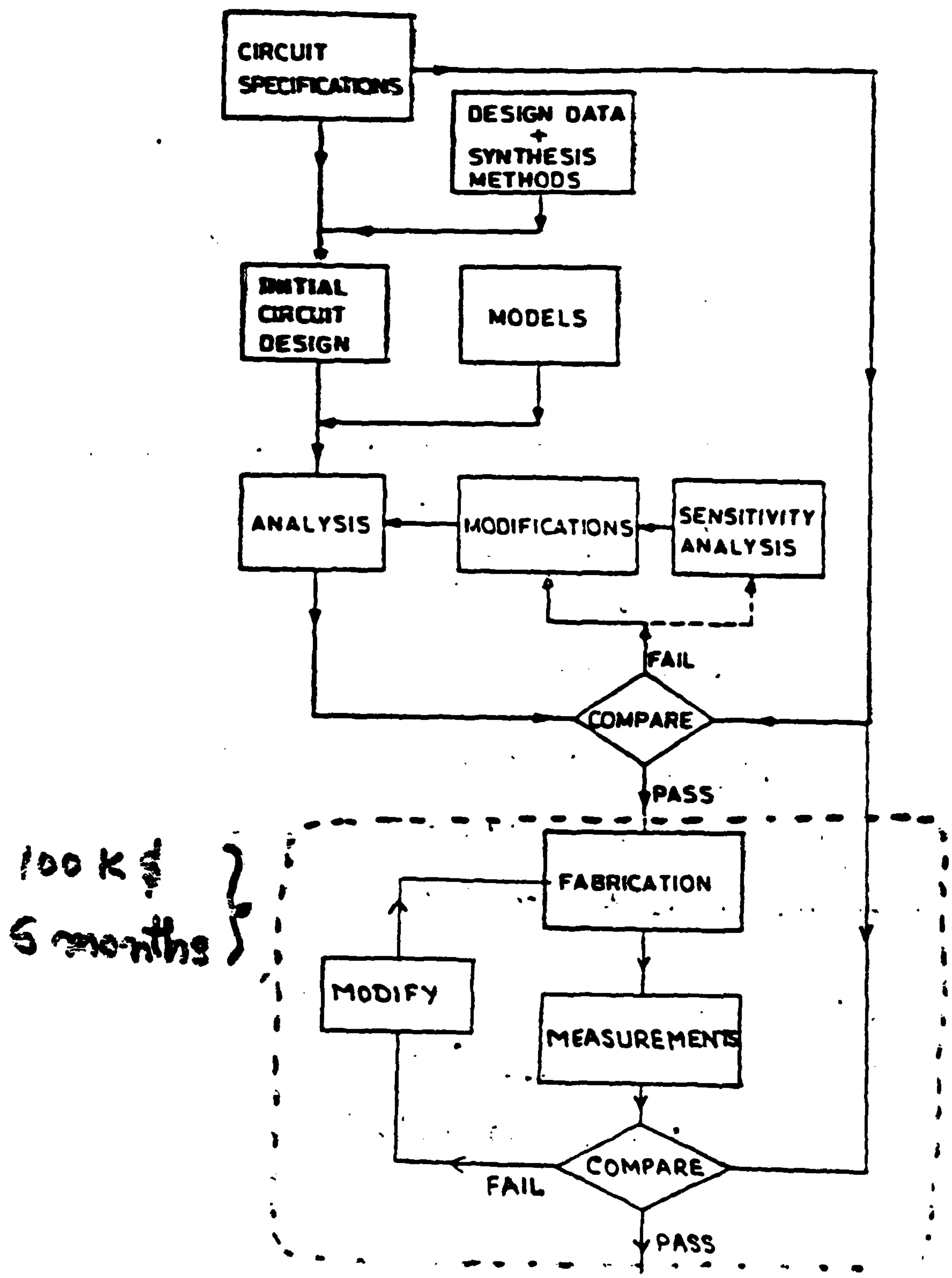
VIEWGRAPH #1



Courtsey: ITT GaAs Tech Center

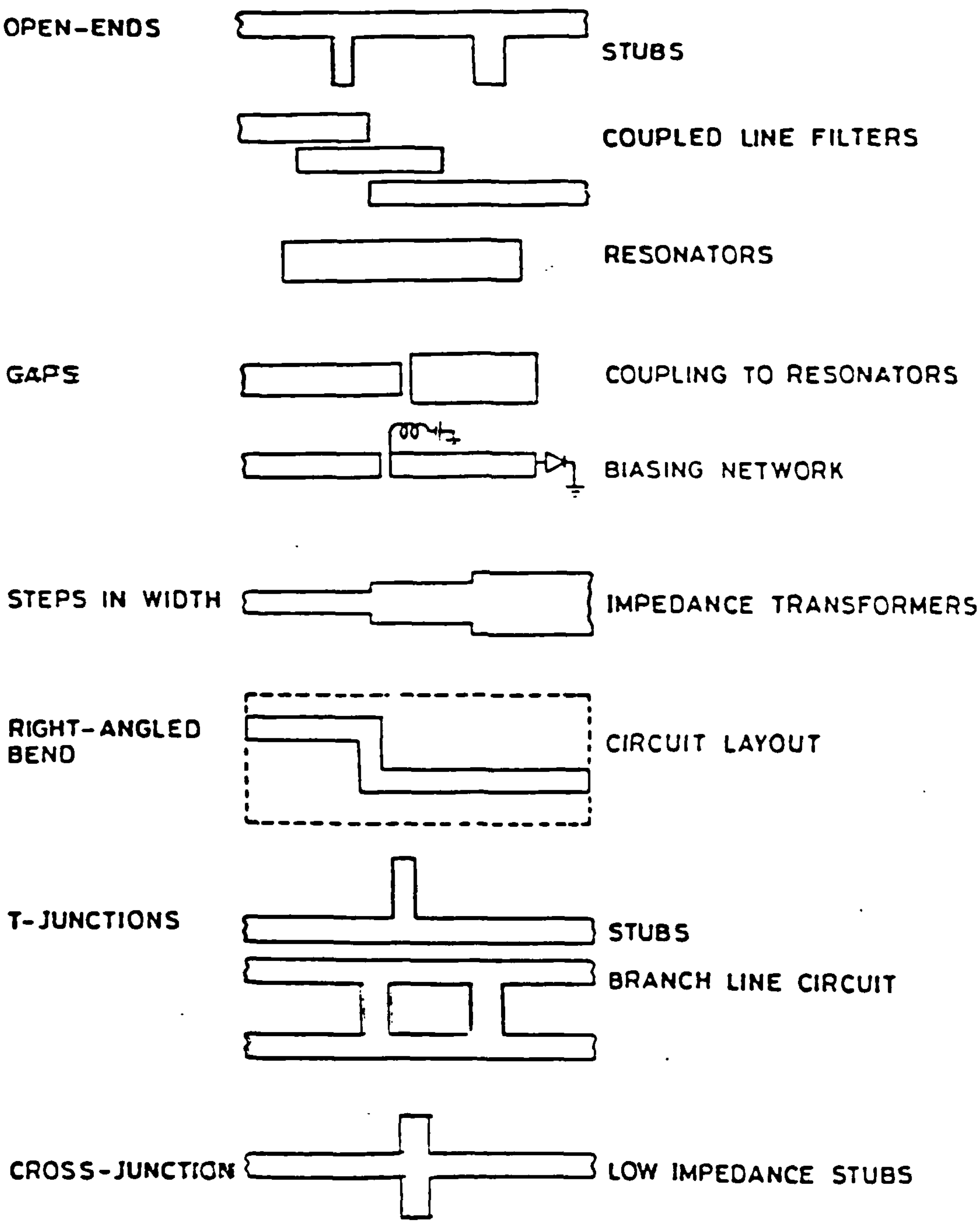
VIEWGRAPH #3

WHAT IS CAD?



VIEWGRAPH #4

3-1



Various types of stripline and microstrip discontinuities and their typical circuit applications.

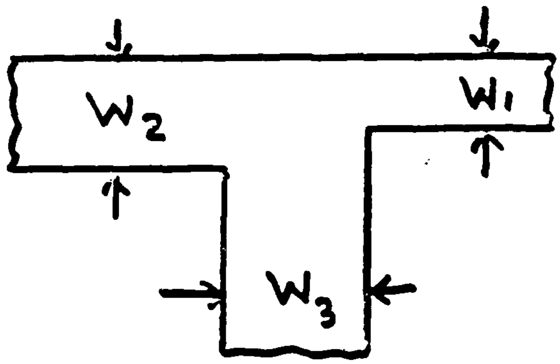
VIEWGRAPH #5

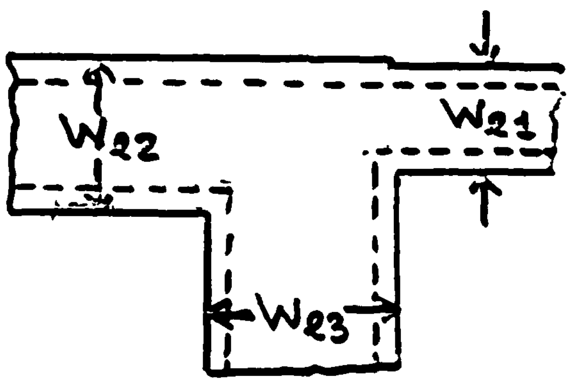
CHARACTERIZATION OF DISCONTINUITIES

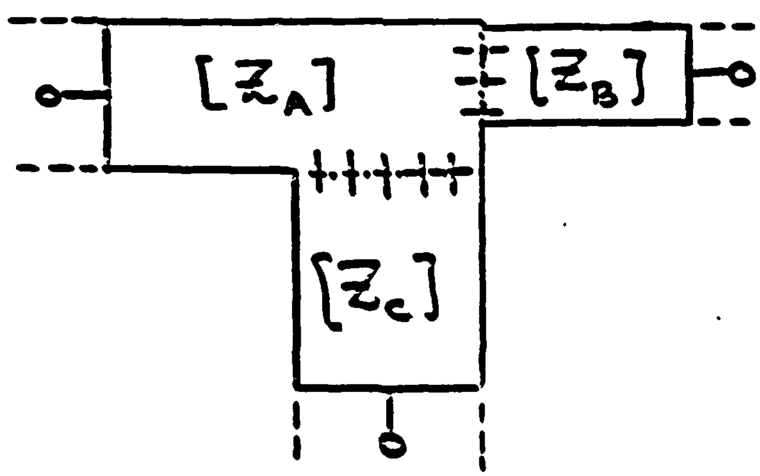
* Quasistatic Analysis (closed-form relations)
* Planar Model Analysis (S-matrix subroutines)
* Fullwave Analysis (numerical results / data files)

VIEWGRAPH #6

Multiport Planar Network Characterization of a Asymetric Tee-Junction

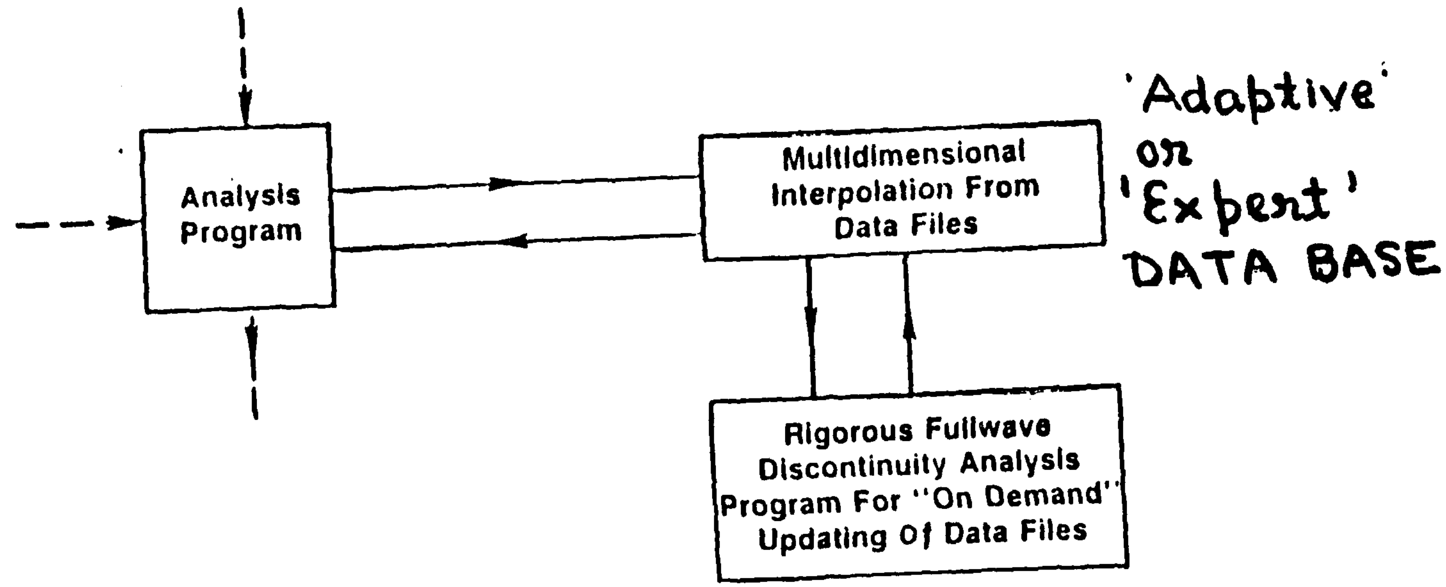






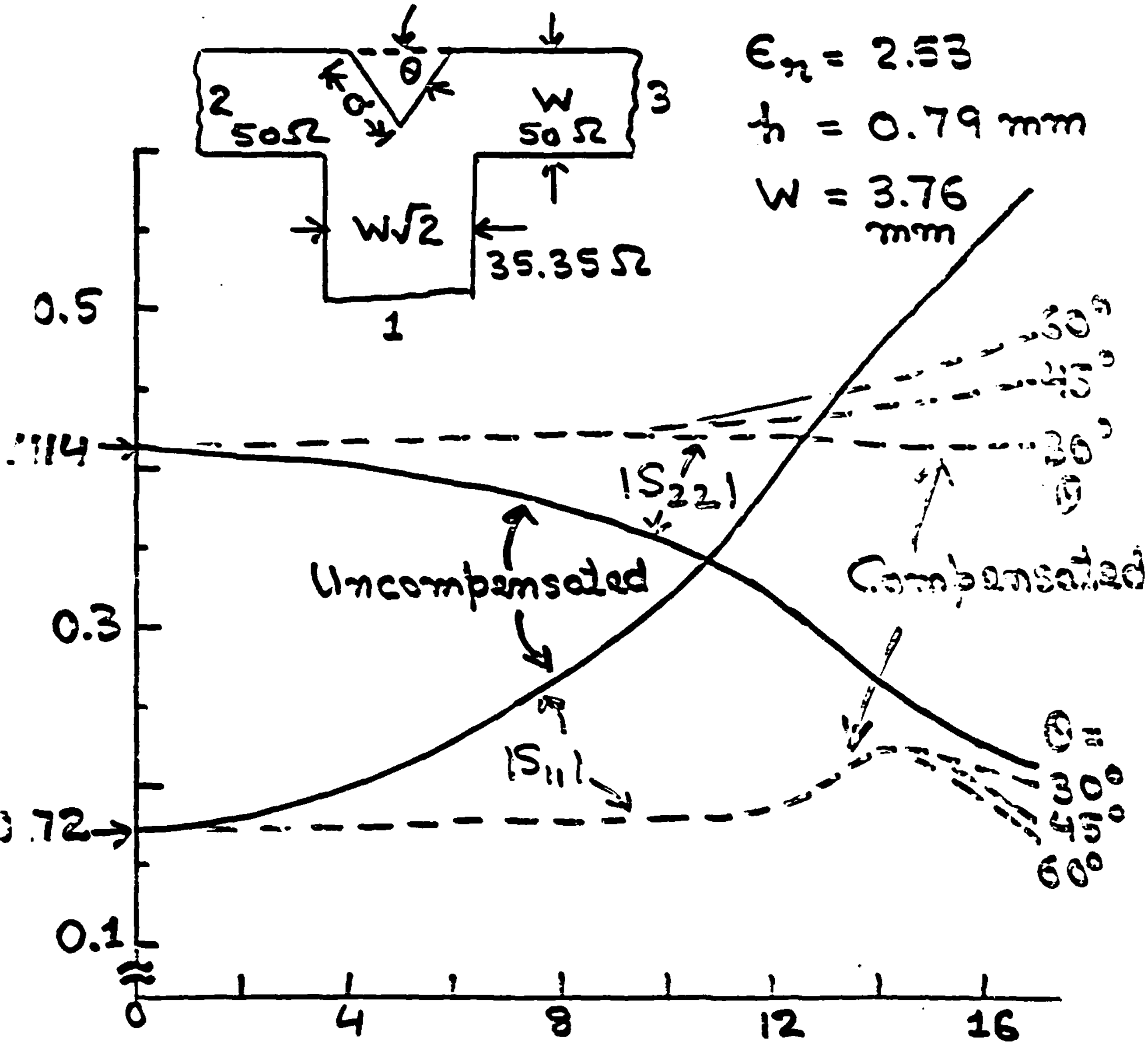
VIEWGRAPH #7

USE OF NUMERICAL MODELS IN CAD



VIEWGRAPH #8

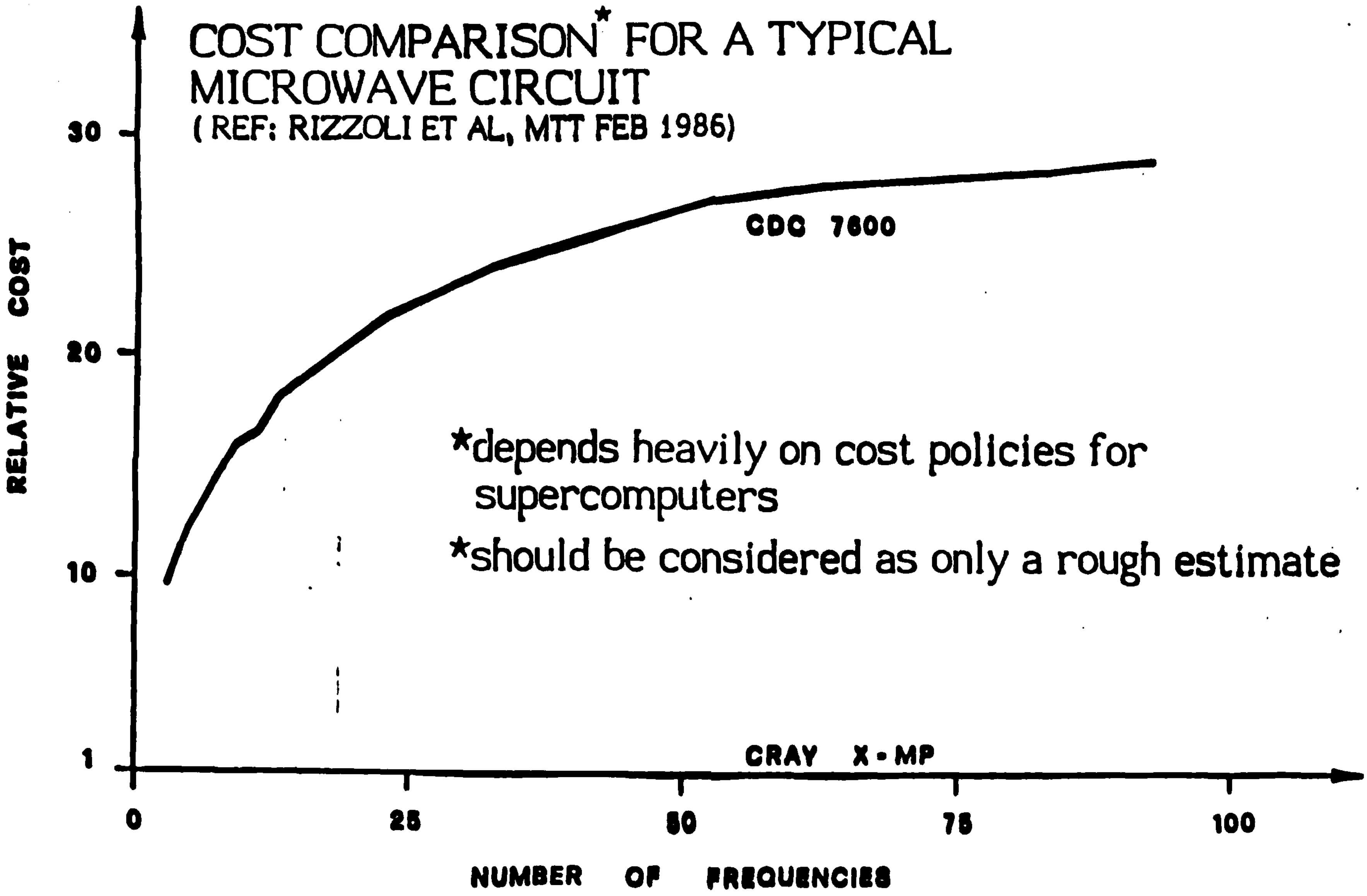
Compensation of a Tee-junction Discontinui



Ref: Chadha & Gupta
MTT-30, Dec 1982

θ	a
30°	0.351 W
45	0.307
60	0.397

VIEWGRAPH #9



VIEWGRAPH #10

We can look forward to:

* Next generation of microwave CAD for monolithic design.

* Use of expert systems / artificial intelligence in microwave/mmwave CAD.

* Extension to millimeterwave frequencies.

* Extension to integrated radiating elements and systems.

* Use of supercomputers.

VIEWGRAPH #11

efficient when the circuit is being analyzed over a wider frequency range. Comparison is shown between scalar processors like CDC 7600 and vector processors like Cray X-MP, and one can see that the cost of using scalar processors is much higher. These results are published by Professor Rizzoli at the University of Bologna. I point out this only to show that this is not a result published by somebody at Cray or some other company who's trying to sell you supercomputers.

I'll conclude by pointing out that we are looking forward to (VIEWGRAPH #11) a new generation of microwave-CAD tools which will lead to accurate designs for monolithic circuits. Use of expert systems (and artificial intelligence to some extent), there is some work already in that direction. A synthesis program which includes some artificial intelligence concepts has been reported recently. There is a need to extend many of these design tools to millimeter wave frequencies, where the main challenge lies in device characterization and modeling. Also, there is a need to extend CAD to subsystems which contain integrated radiating elements (microstrip antennas). Finally, use of supercomputers will make microwave-millimeter wave CAD more efficient and accurate. Thank you, these are my comments. Any questions, comments ...?

CHETHIK: I have an observation which stems from perhaps a rather incomplete understanding of this subject. It seems to me that much of the design of microwave integrated circuits involves stripline techniques and transversal elements, and I'm wondering if this is a tradition born out of stripline hybrid circuits that have been translated into this new medium, like gallium arsenide, and has there been an attempt to your knowledge to use lump element designs with small dimension components in GaAs? I would think that the design and analysis may be simpler perhaps with components whose dimensions are a small fraction of a wavelength. Could you comment on that?

GUPTA: Yes, I have two comments on that. First of all, lumped elements have been used at microwave frequencies. They have been used at 12 gigahertz and to a limited extent up to 18 gigahertz. They could perhaps be extended to higher frequencies by making them smaller initially. But there are two difficulties that come. Lumped elements at microwave frequencies are smaller than the distributed elements. So you end up with lower Q-factors and higher losses. That's one of the operations that's limiting the use of these lumped elements to higher and higher frequencies.

The second operation is that at microwave-frequencies it's not easier to design lumped elements. In fact today, the hardest thing to design for a microwave designer is to design some of the passive lumped elements that one wants to use. Our design of microstrip lines, an analysis of these transmission lines, is far more advanced than the design of a simple inductor at 15 gigahertz. Stray capacitances cause self-resonances in lumped inductors at microwave frequencies. So these have not been used commonly for the design at higher microwave frequencies. Any other comments?

SCHOLTZ: I can appreciate that your problems become higher dimensional than the circuit design at lower frequencies, but I'm struck by the fact that EESof's touchtone program, when I was talking to Charles McGuire who unfortunately hurt his back I guess cleaning his garage this weekend and couldn't make it, as I under-

stand it runs on a PC and works up I guess to the point where chip size becomes significant relative to wavelength, and the fact that you need a supercomputer to do this job, is there any hope that this task will eventually come down to a more reasonably sized computational machine?

GUPTA: Yes, I'm familiar with touchtone -- I have a copy of that. Also, there are other programs which run on PC and do a reasonable analysis. One of the main difficulties in most of these programs is the limited accuracy for the characterization of various components that go in monolithic millimeter wave circuits. The complexities of codes increase when you go to higher frequencies. Some more detailed modeling could be done on supercomputers. You want to do that because it's cheaper to carry out detailed simulation on supercomputers than on smaller computers. What I see is that many of the models that come out of these detailed simulations of the supercomputers would eventually get coupled to a workstation type environment, which is used in carrying out day to day design. Any further comments?

HUTH: You were talking about supercomputers -- what kind of supercomputers are you talking about, Crays, are you talking about distributed multiple processors?

GUPTA: The results I showed were on Cray and for our work we use a Cyber 205 because it's more easily accessible to us; it's not for any other reason.

HUTH: Some of the outsiders showed me some things about doing equivalent things with distributed superminis, is basically what they're talking about, because the price was so much cheaper on the superminis. You can take 10 superminis and get almost the Cray performance for about a tenth of the price.

GUPTA: That's really true. Also, I am not trying to suggest that in order to use these CAD tools you need to own a Cray.

HUTH: Yes, right. I guess that was the point I was trying to make, is that there are people that are talking about, the point is there's a lot of computations that have to be done. Now the question is do you distribute it among multiple processors or do you put it on a Cray, and either way whichever you've got available to you?

GUPTA: Supercomputers are ideally suited when there are similarly structured computational steps in algorithms. This is true for microwave circuits since there are similar computations to be done for the various parts of the circuit.

HUTH: Exactly, so it lends itself to parallel processing, that is what I was trying to get across.

CHOMA: Thank you, Prof. Gupta. As Bob Scholtz just mentioned, I regret to say that Charles McGuire is not here with us owing to a back injury. If you have a pencil and paper, though, Charles McGuire I'm sure is very willing to discuss the touchtone and all of the other software that he's working on. The number he can be reached at, or you can arrange an appointment with or whatever, is (818) 991-7530. I'll repeat that, this is for Charles McGuire of EESof, (818) 991-7530. He was trying to be a good guy and cleaned out his garage, which we were smart enough not to do over the weekend. Anyway, on behalf of Charles I'd like to apologize to this group for his inability to attend, but as I say I'm sure he'll be more than willing to discuss his work with you. Our next speaker for the afternoon therefore will be Randy Reitmeyer of the U.S. Army Labcom in Fort Manmouth, New Jersey.

Randy ....

REITMEYER: Okay, it's a pleasure to be here. I guess at one level some of you may look at me and say there's a customer, and I hope to start off with a little bit of controversy by saying in effect that we in the Army as a customer need a whole new way of doing business, in terms of acquiring, developing, and life-cycle managing systems, and that what I hope to transmit today is that we have a crusade in our midst; that we are pushing forward to do just that. In fact, to accomplish that we certainly can't do it ourselves. We're thrilled to be here, myself and Charlie Bosco, from the Army, and of course Bill Sander, from ARO, because we can't do it alone. We don't have enough Army dollars to do what has to be done. There's a lot of research opportunities that I hope to reveal by some of the things that we're already doing, and we're excited about some of these things. Basically, I would have to say that many of the problems that we have today -- or opportunities -- are really seeded at the system level. I can say that even though I've been in the microelectronics component business for about 20 years, this is where I believe the focus needs to be. The theme that I'm going to be presenting is basically designing electronic systems that are affordable and supportable. As way of background, for those of you who were around in the microelectronics industry when LSI became popular, we in the Army desperately were trying to find ways to using that technology in our systems.

We did a survey in the early 70's to find out who could design the integrated circuits that were rapidly growing in complexity, and we found out that, by and large, not too many companies had the design experts, nor the CAD tools, that would give us rapid turnaround and low cost, and of course we wanted first-pass success on silicon. Therefore, we and some of the other agencies launched the development of tools primarily based on standard cells, and gate array design methodologies for low cost and rapid turnaround. The chip level design problems generally have been solved by a whole CAD industry that has spawned today. We've basically gotten out of that business in terms of computer-aided design. We've gone back and looked at the problems that are there today, and instead of complex chips being the problem, it's now complex systems that we see as the problem, with a lot of problems surrounding that. We ask the question, "Are there enough really good system engineers and do they have the tools that they need to rapidly and affordably provide the Army, Navy and the Air Force with the systems that we need?" You've seen this type of chart that shows the growth in system complexity the Army has fielded. One of the problems that we see is the system engineering problem of being able to rapidly design complex systems. We definitely perceive this to be a real issue.

With complexity comes a major problem for the Army -- and that is, we are a paper-based operation. We capture information on paper, which is not very worthwhile for us. It's good information if you want to build copies of a system. But, if you want to change a system at some later point in time -- if there is a threat change, if there is a maintenance problem -- if you want to change the system for any reason, paper design documentation is not the way to go. Yet, that's the way my company does business right now. We defy anyone to read that documentation -- (FIGURE 3) by the way, this is documentation for a VHSIC chip -- and do anything

useful with it. So, in terms of a new way of doing business, it is obvious that we must find new ways of capturing design information. That's one of the things I'll get to in a moment.

For our company, growth in complexity translates into cost (FIGURE 4). The cost of design, document, maintain, upgrade ... all are escalating tremendously. And, as you all know, military budgets are not growing in the same direction as the costs. So, another major problem that we collectively have to solve is how to produce an affordable design and manage it.

Our organization, for many years, has been developing methodologies and tools for the design of chips. When you look at the cost for Army systems -- the cost for the hardware and maintenance over a life cycle of 25-40 years -- most of the costs are at the back end (FIGURE 5). So, our investment strategy is only 10% of the problem. The message here is that we shifted our strategy about 3 or 4 years ago to develop methodologies and tools to deal with the whole life cycle.

It takes the Army about 10 years to field a system (FIGURE 6). Unfortunately, when a new system is being fielded, many of the microelectronic components are obsolete technology. We have to be able to deal with that problem. So again, we need methodologies and tools that will allow us to deal with that problem. What I'm doing in this talk is presenting a variety of problems because some people are interested in one kind of problem, and others in different problems. Then I'll show you some examples of how perhaps we can work together.

This is the M-1 tank (FIGURE 7) and this is a rack of test equipment and the maintenance manuals to go with it. The interesting thing about this is that all but one of those boxes are full of cables. The reason there are so many boxes of cables is because a common buss was not designed into the system when the system was being engineered. Therefore, the Army has to lug along all of these boxes even though it should be fighting a new way in the future ... the Army is supposed to be mobile ... it's not supposed to be lugging around a great deal of test equipment. [laughter]

So, the methodology that we must have -- someone asked a question about testability -- also has to deal with the testability issue. This block diagram depicts the complexity of what goes into the LHX. Tremendous processor capabilities are required for a single pilot. These present tremendous system engineering problems. We need CAD tools that can deal with these needs. The last point I want to make, before I get into some of the solutions, is that we're looking for a streamlined acquisition cycle that fields systems in 4 to 5 years ... instead of 8 to 12 years. So, time spent on the several outlined problem areas such as complexity, parts obsolescence, testability and other issues will have to be cut in half, if you can do this. Something has got to give, there has to be a new way of doing business. It's got to be a computer aided engineering process. Some of the ideas that we and the services are working on is a different strategy that is based on modeling the entire weapon system. The research opportunities there are for that have got to be tremendous. You have to be able to optimize the system design, the hardware and software, the testware and anything else that goes into a system. We believe that there is a tremendous need for automating the hardware design through knowledge based compilers.

As I said before, we've got to be able to simulate the total system design prior to

building. What we would like to see is for a contractor to be able to come in, sit down and load a design database and go through a design review in a computer instead of using just viewgraphs or paper. We want to actually be able to analyze and execute that design model. We've got to be able to assess in that model its testability, reliability, and especially, affordability. Most importantly, we have got to be able to archive the total design of the system ... not just the schematics; not just the parts list. We've got to be able to archive the total design, so that any contractor at any point in time during the life of that system can upgrade it, can value engineer it, to improve the cost. We can't be locked into technology and contractors.

The starting point of this change in the way business has to be conducted is to adopt a structured, hierarchical methodology for designing systems. Although you can't see this (FIGURE 11), what it says is that we have to be able to describe our system functionally and structurally from the top down; capture that information and utilize that as the life of the system goes on. Our selection for the first CAD tool able to do this is the VHSIC Hardware Description Language (VHDL). For those of you who have watched what's been going on in VHSIC programs, we've invested about $15 million in the development of VHDL and a tool environment to go with it. I'll show you some examples of how we're starting to utilize that concept already. For those of you who are not experts on hardware chip design or who may not even be familiar with it, I will at least say this much ... VHDL, at a minimum, has the capabilities of describing hardware two critically different ways (FIGURE 13). First, you can describe functionality, and on this level, we functionally describe hardware in a way that is independent of technology. Secondly, we can use the language to describe hardware structurally. With one common language we are hoping to capture the design through all levels (i.e., system, to the board, to the chips, to the macro).

We've got a lot of converting to do within our own company. We've talked about these ideas to 4-star generals and other technical people in the Army -- yet to many people, it doesn't register. They've never heard of *disciplined hierarchical* methodology. So, what we're doing and how we link back into the communications community is that we've teamed up with the Communications-Electronics Command (CECOM) of Fort Monmouth to re-engineer a multiplexer in the Army inventory called a TD-660 (FIGURE 14). They brought this system to us and said, "Won't you please solve the TD-660 parts obsolescence problems?" Rather than just replace one integrated circuit with another integrated circuit, we said, "Allow us to use the structured methodology and the hardware description language on the system." So, we went about capturing this system -- first as a black box, functionally and structurally; then captured the design of that system in as much hierarchy as we could. We then re-engineered the guts of that system using today's gate array technology, and, through the miracle of microelectronics, as you're all familiar with, the printed circuit boards in the system that look basically like these at the top (FIGURE 15) were re-engineered as shown below. Six boards are populated and four are now blank. The four blank ones are there just to provide system connectivity.

In the re-engineering process, the system has greatly improved the mean time between failure (estimated to be at least 2 to 3 times better than it was before) and

power consumption is down by 33%. We're going to save a bundle ... about $1.1 million per year on the maintenance.

The most important thing that we think we've transmitted to people in the Army is that we have to do design for all times and archive that (FIGURE 17). I don't think there's another military system that has included this yet. So, there's a first example where we're trying to bring about a revolutionary way of doing business by using the VHSIC Hardware Description Language. We've done a similar kind of thing for the PRC-70 radio, which again, is a communications system for the Army. What we've demonstrated is that we could re-engineer chips by capturing the chip design and emulating the function of the old chips with gate arrays.

Overall, what these examples say -- and this has already been said by previous speakers -- is that what it takes is a very powerful design system (FIGURE 19). We believe it is going to require a considerable amount of expert system/AI type capabilities, a major database and engineering information system because there are not enough designers to go around. We would like rapid prototyping; we need a lot of design synthesis. We need a very powerful assessment tool to take care of reliability, affordability, and a lot of other "-ilities" which have to be assessed. You've got to be able to do total hierarchical simulation. You've got to be able to do very powerful and fast cost modeling. That all has to be embodied in a CAD system.

Everybody has a design process diagram (FIGURE 20), and we've got one too. The point I wish to emphasize is that not only are we using the VHSIC Hardware Description Language, but we are also using other tools, which are coming out of the VHSIC program, at the higher levels as we move towards the system level.

We are currently investing in a tool -- developed by RTI -- called the Architecture Design and Assessment System (ADAS). The so-called ADAS toolset (FIGURE 21) has the capability to do color graphical based co-design of both software and hardware. This capability allows you to capture both hardware and software kinds of things and makes trade-offs at the architectural level and then be able to map software onto the hardware. I'm not going to go into detail and that's because I'm not an expert on ADAS. It basically makes use of the directed graph methodology using PETRI NET techniques. We have developed interfaces from ADA to the VHDL, as well as the silicon compiler. When you look at this kind of capability, we feel that we've got the beginnings of a scheme that could take us all the way from architectural level with ADAS, down to the detailed functional/structural design using VHDL. So you see the beginning of some rapid prototyping capability.

One of the special emphases that we have in the Army -- the question was asked about testability, and you saw the picture I used of the M-1 tank -- we've been asked to join forces with one of the major portions of the Army to look at the problem of designing testable systems. So, one of the things that we're doing with VHSIC is to take the ADAS capability and use that as a method to first capture the requirements of the system at the architectural level, and then create a new capability called TEA (FIGURE 22), which is the acronym for the Test Engineers Assistant, used to test the engineering of the system. Now what TEA is to provide for us is the capability to aid the designer by building testability in at the system level. By having design for testabil-

ELECTRONICS TECHNOLOGY AND DEVICES LABORATORY

DESIGNING ELECTRONIC SYSTEMS THAT ARE AFFORDABLE and SUPPORTABLE

R. A. REITMEYER

ELECTRONICS TECHNOLOGY and DEVICES LABORATORY

FT. MONMOUTH, N.J.

AVN: 994-3465

LABCOM - Providing Soldiers the Decisive Edge

U.S. ARMY
LABORATORY COMMAND

FIGURE 1

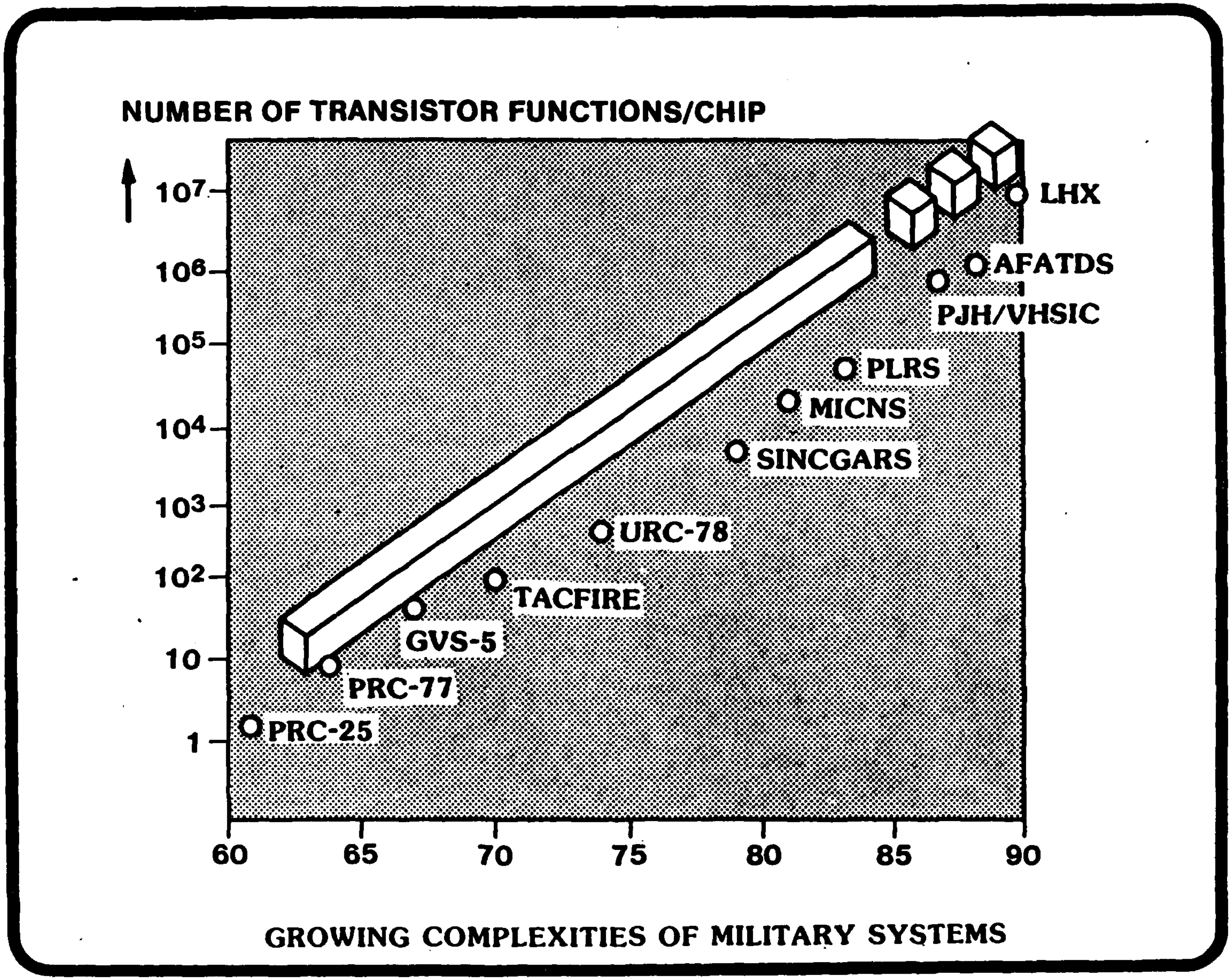


FIGURE 2

HISA-FM 687-78

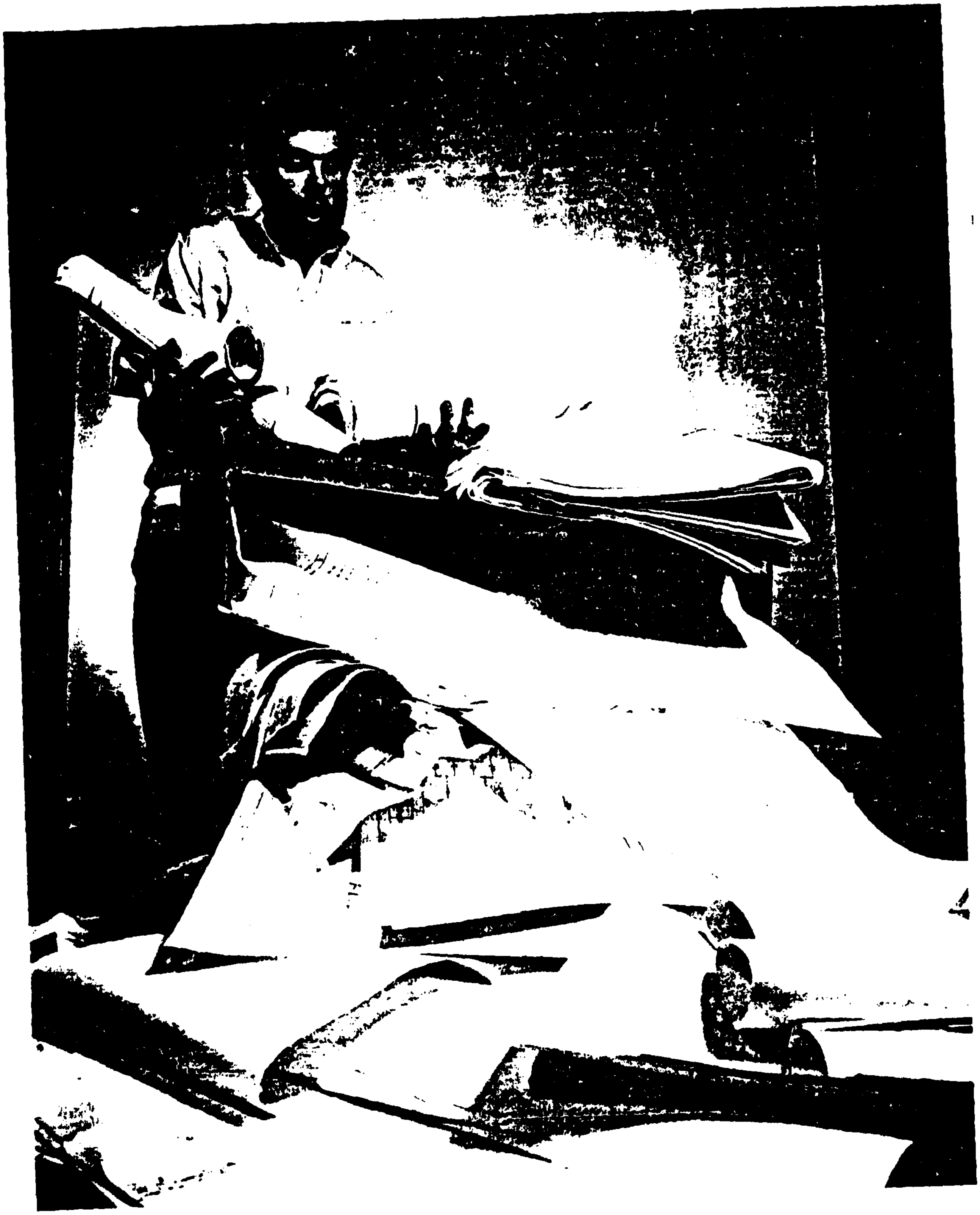


FIGURE 3

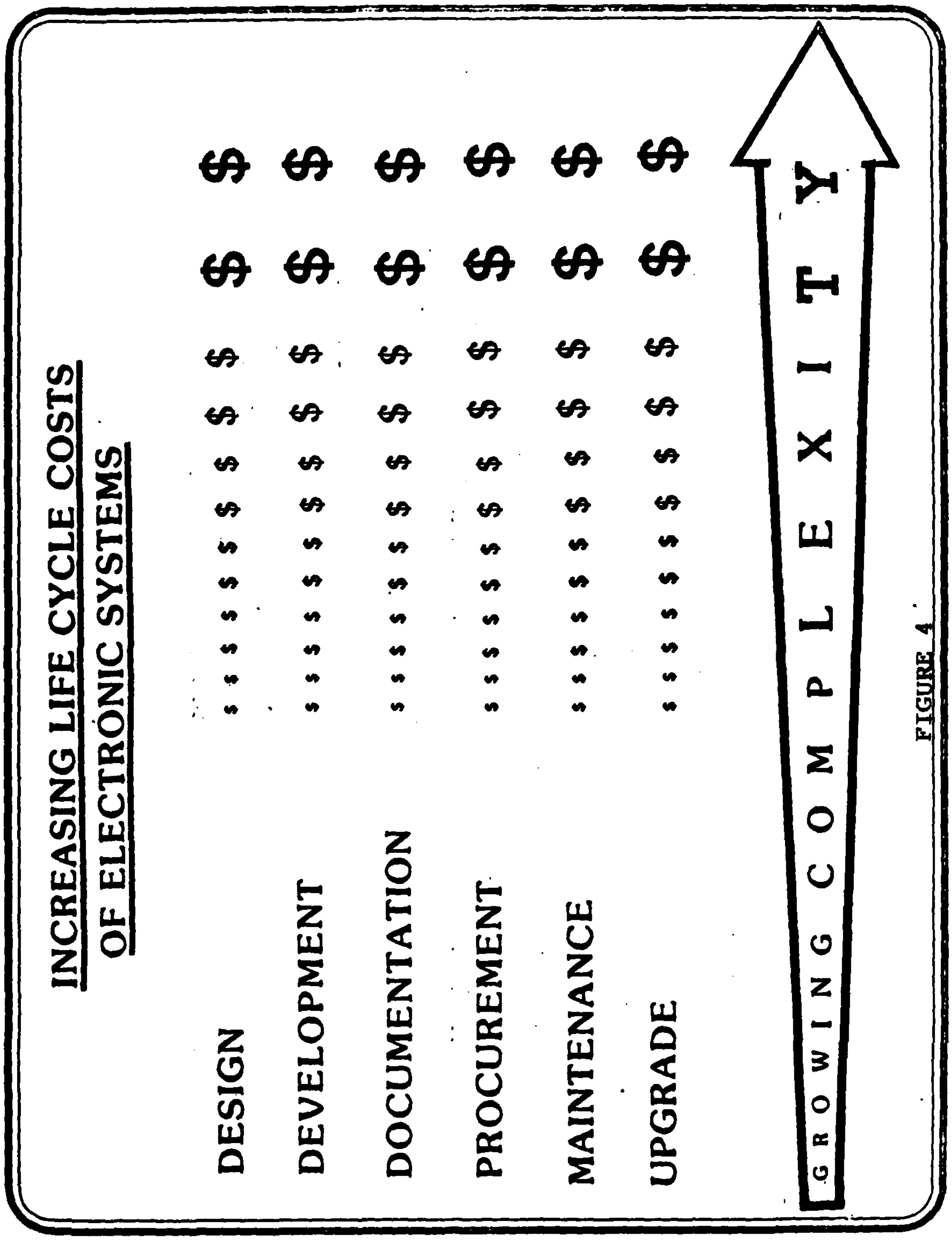

INCREASING LIFE CYCLE COSTS
OF ELECTRONIC SYSTEMS
DESIGN
DEVELOPMENT
DOCUMENTATION
PROCUREMENT
MAINTENANCE
UPGRADE
G R O W I N G C O M P L E X I T Y

FIGURE 4

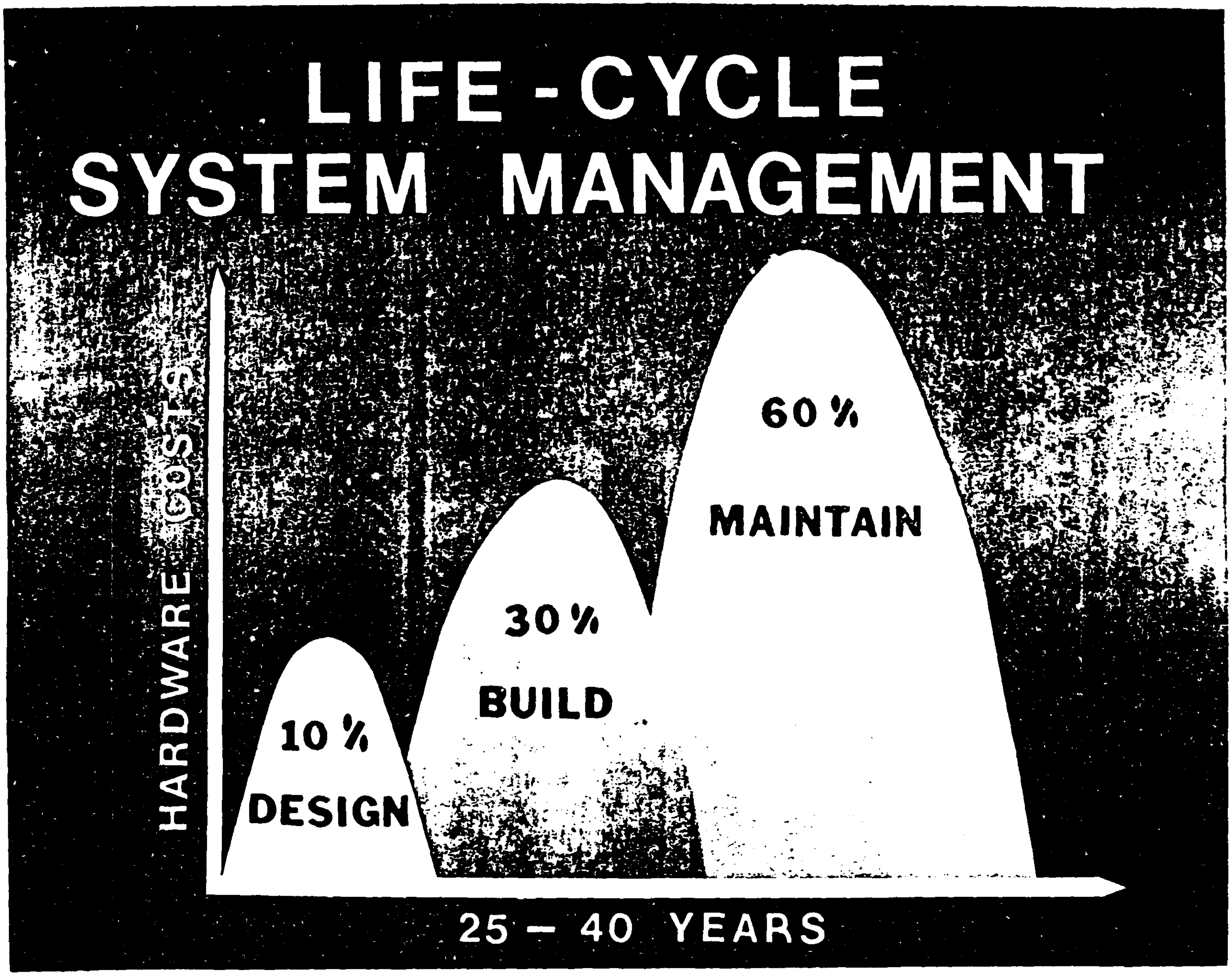

LIFE-CYCLE
SYSTEM MANAGEMENT
HARDWARE COSTS
10 %
DESIGN
30 %
BUILD
60 %
MAINTAIN
25 – 40 YEARS

FIGURE 5

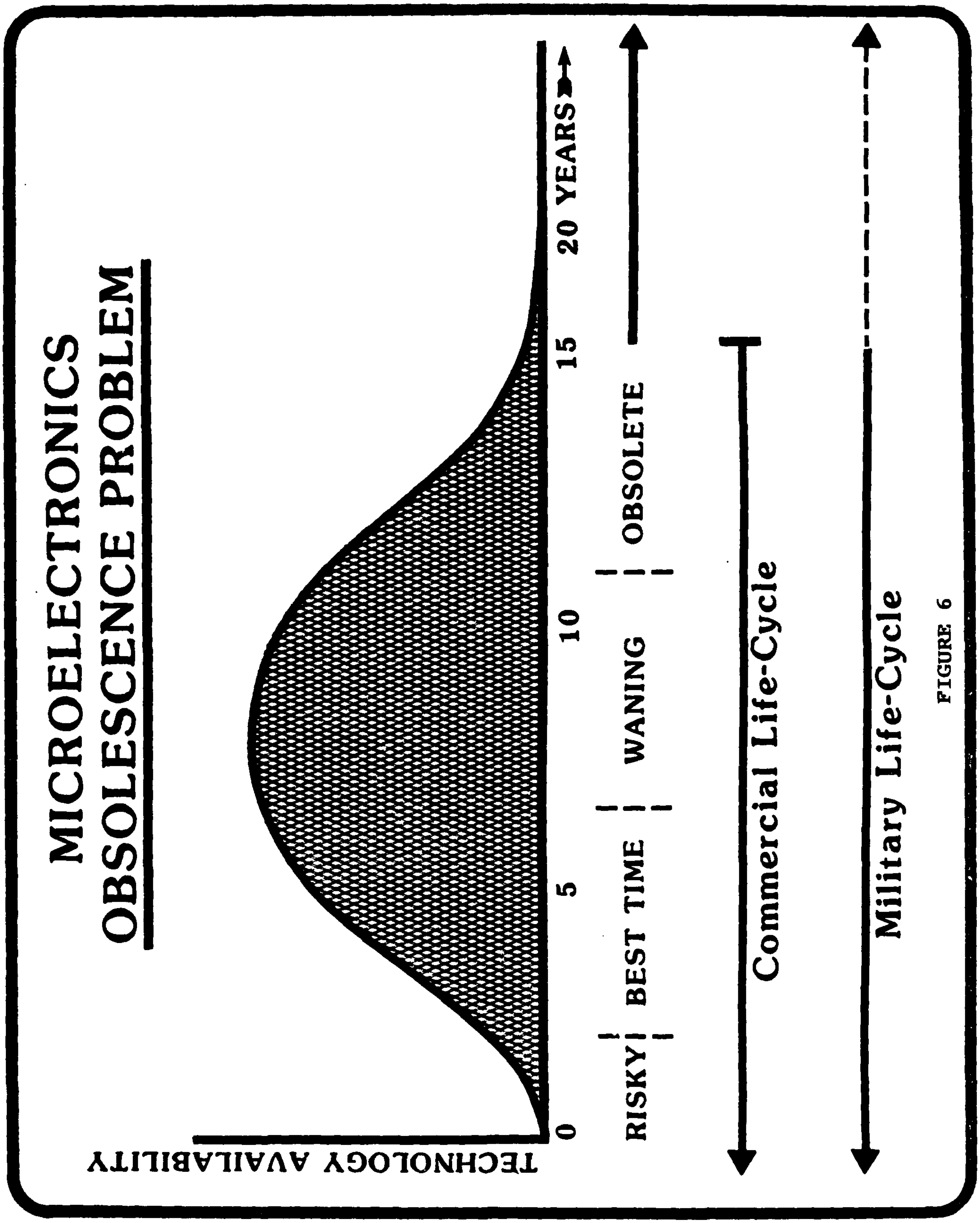

MICROELECTRONICS
OBSOLESCENCE PROBLEM
TECHNOLOGY AVAILABILITY
0
5
10
15
20 YEARS
RISKY
BEST TIME
WANING
OBSOLETE
Commercial Life-Cycle
Military Life-Cycle

FIGURE 6

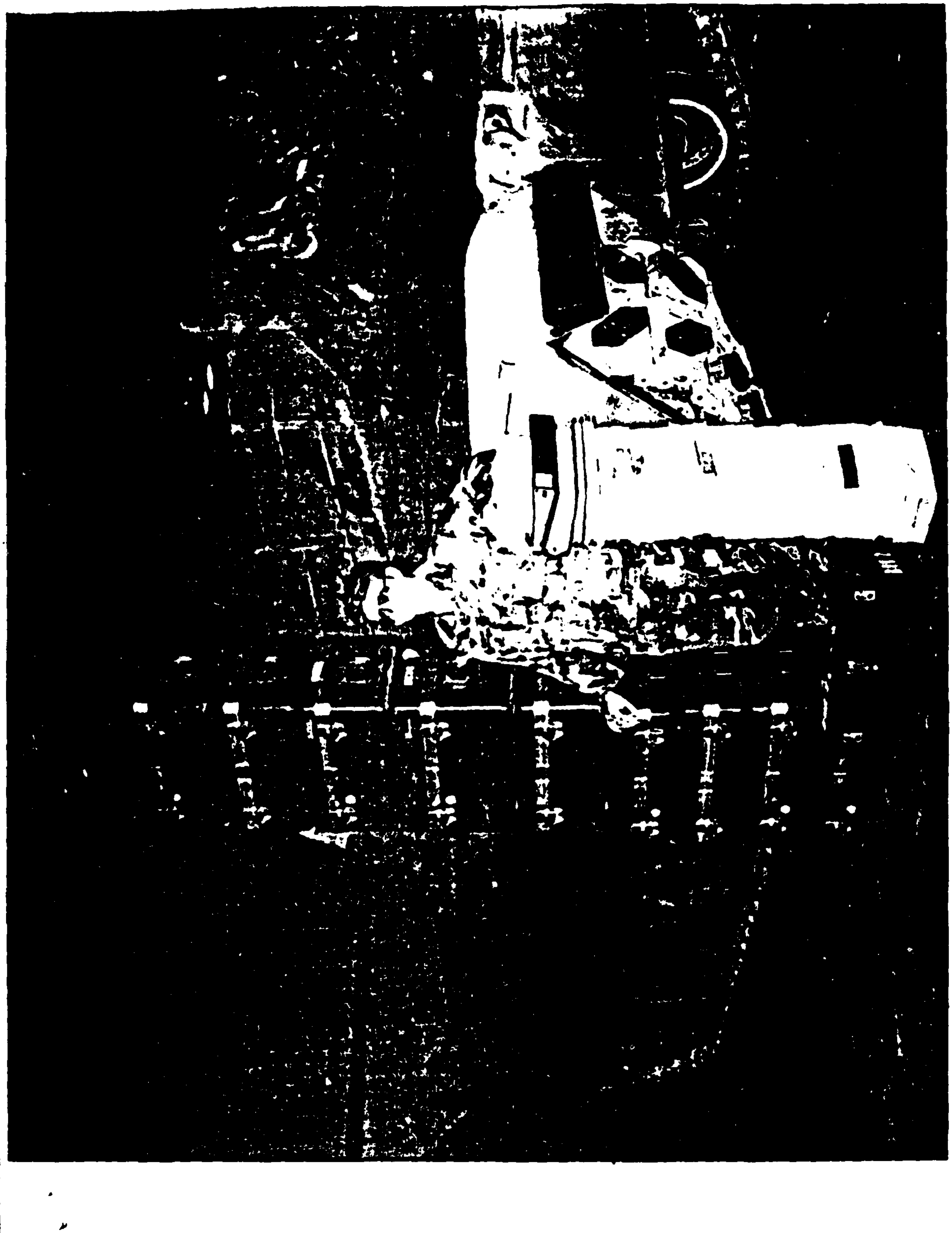


FIGURE 7

LHX INTEGRATED AVIONICS

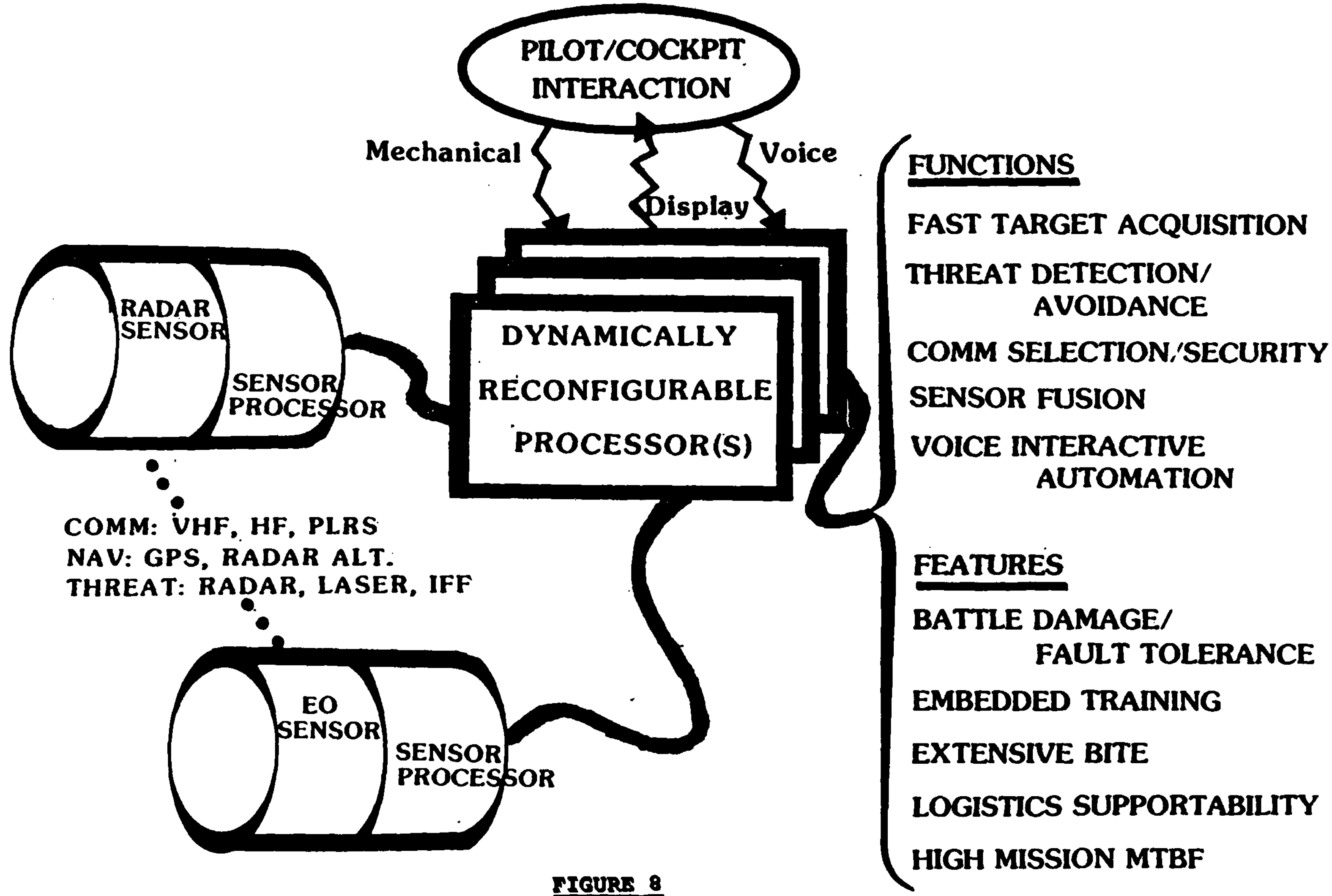


FIGURE 8

CURRENT ACQUISITION CYCLE

8 - 10 YEARS

STREAMLINED ACQUISITION CYCLE

4 - 5 YEARS

LABCOM - Providing Soldiers the Decisive Edge

U.S. ARMY
LABORATORY COMMAND

FIGURE 9

STRATEGY

1. MODEL AND OPTIMIZE SYSTEM DESIGN/SOFTWARE FOR PERFORMANCE, TESTABILITY, RELIABILITY, ETC.

2. AUTOMATE HARDWARE DESIGN WITH KNOWLEDGE-BASED COMPILERS.

3. SIMULATE TOTAL ELECTRONIC SYSTEM DESIGN PRIOR TO BUILDING.

4. ASSESS TESTABILITY, RELIABILITY, MAINTAINABILITY, AFFORDABILITY.

5. ARCHIVE TOTAL SYSTEM DESIGN FOR FUTURE TECHNOLOGY INSERTION.

FIGURE 10

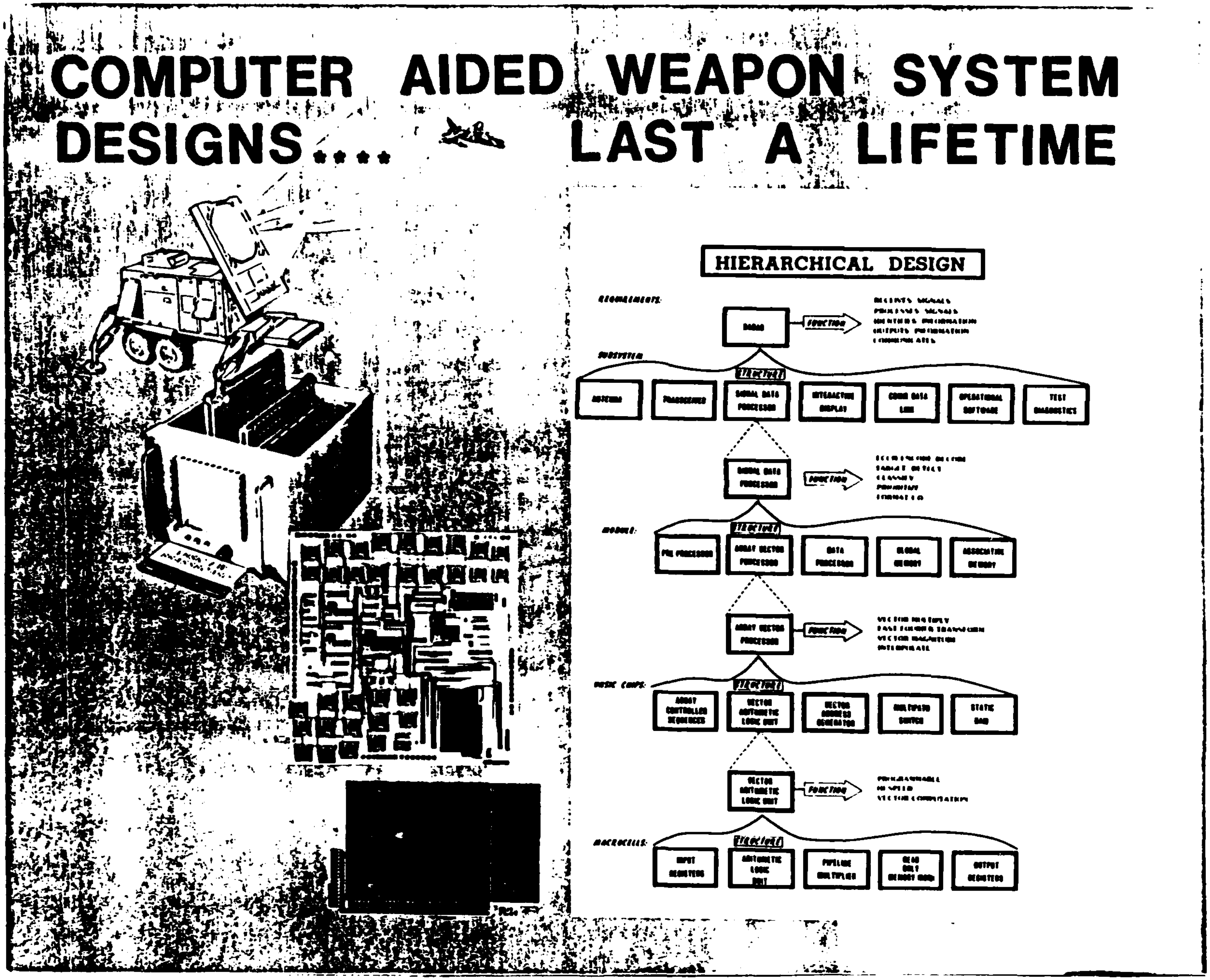

COMPUTER AIDED WEAPON SYSTEM
DESIGNS.... LAST A LIFETIME
HIERARCHICAL DESIGN

FIGURE 11

VHSIC
HARDWARE DESCRIPTION LANGUAGE

- FOR SUCCESSFUL HIERARCHICAL DESIGN
- ALL MODULES DESCRIBED IN A COMMON LANGUAGE (IN ADA SOFTWARE)
- TOTAL SYSTEM SIMULATION

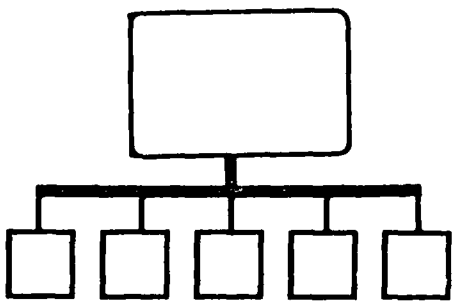
FIGURE 12

HARDWARE DESCRIPTION LANGUAGE (HDL) IS:

- FUNCTIONAL DESCRIPTIONS

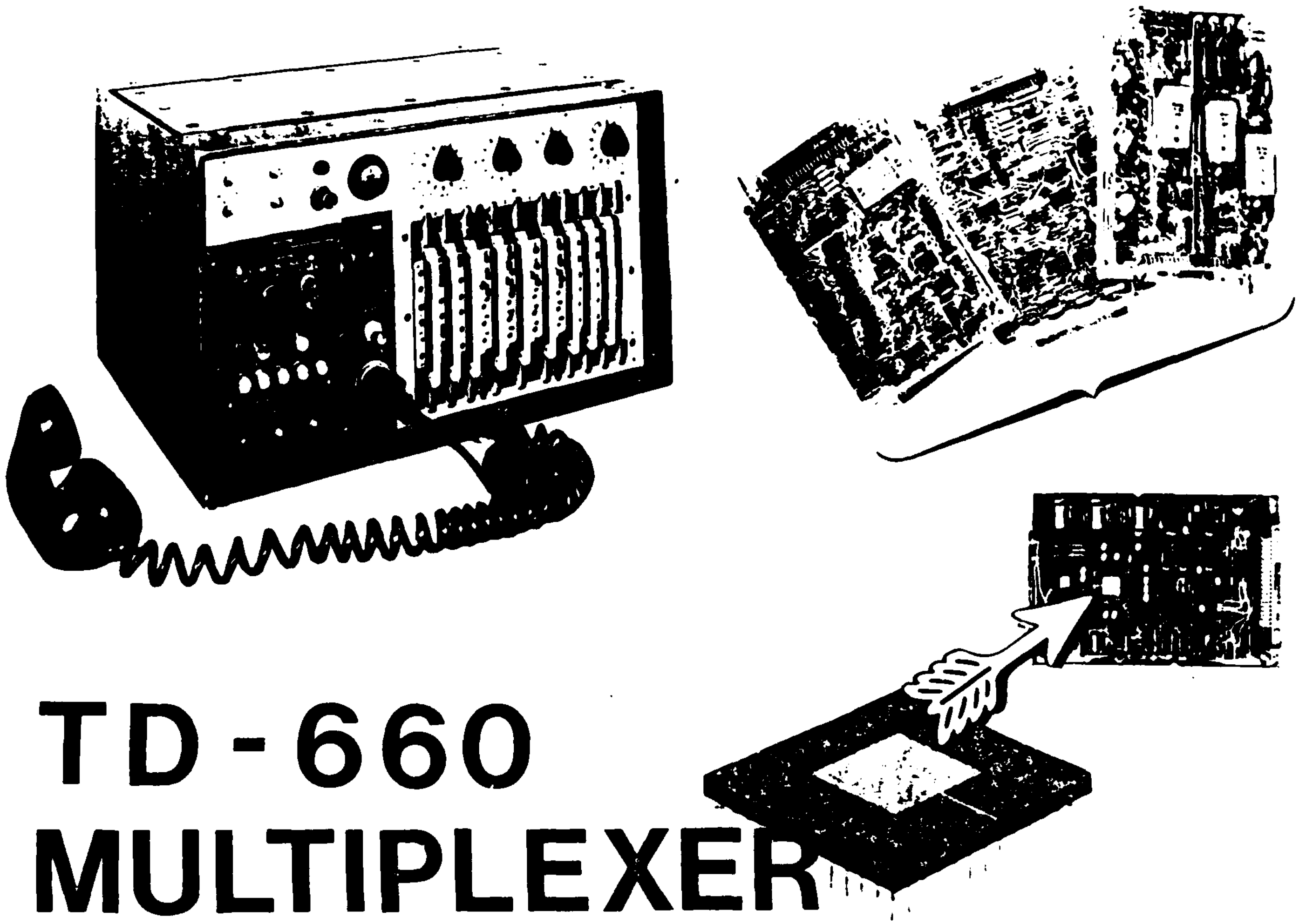
WHAT IT DOES

- STRUCTURAL DESCRIPTIONS



HOW IT DOES IT

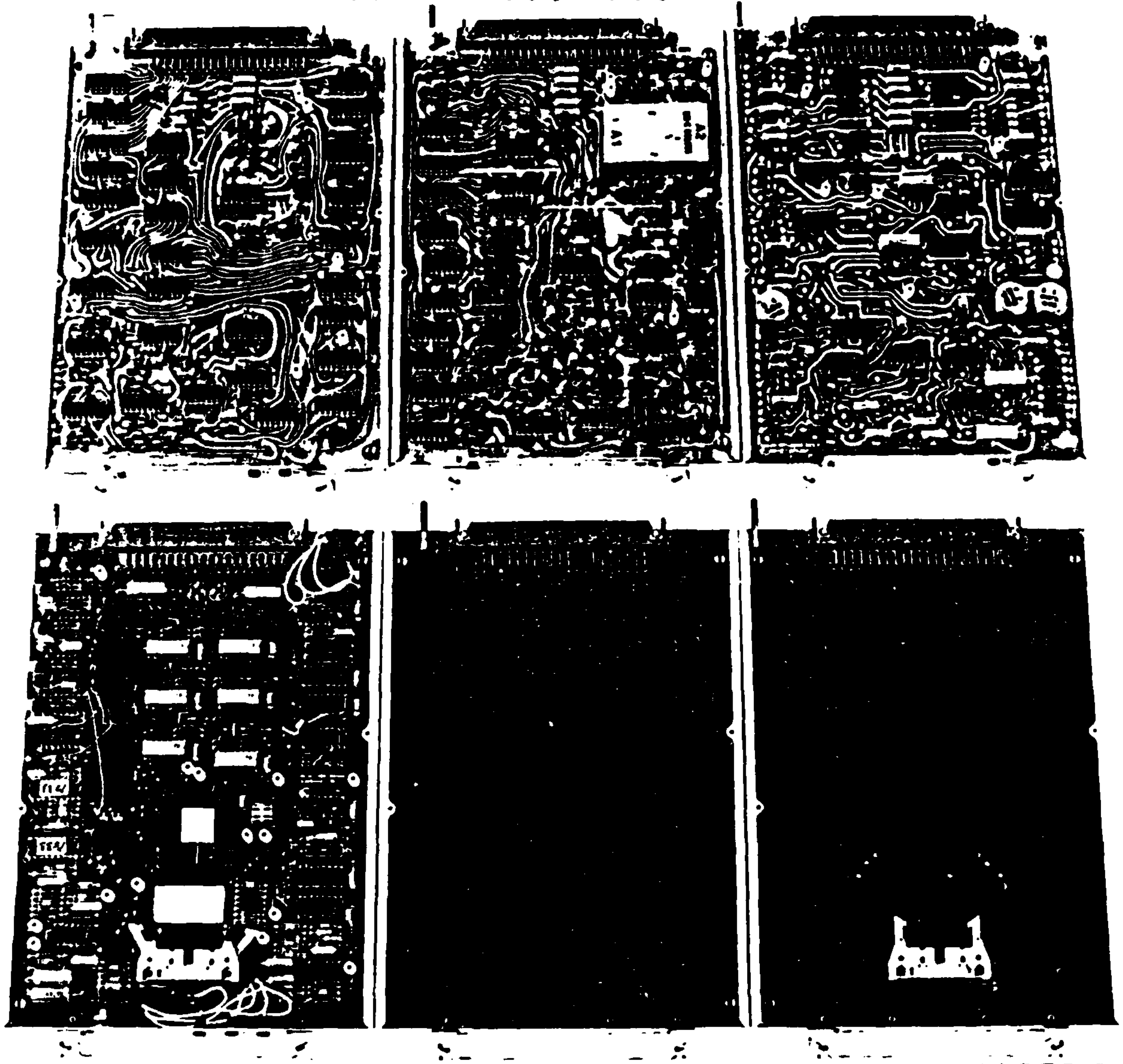
FIGURE 13



TD-660 MULTIPLEXER PARTS EMULATION

FIGURE 14

TD-660 HARDWARE



NEW BOARDS △ EMULATE FUNCTIONALITY & REPLACE OBSOLETE PARTS

FIGURE 15

TD-660 PROGRAM BENEFITS

MTBF - > 2:1 IMPROVEMENT	POTENTIAL FOR $ 1.1M SAVINGS PER YEAR AFTER 4 YEARS
POWER CONSUMPTION ↓ BY 33%	DOCUMENTED IN HDL FOR FUTURE CHANGE AND COMPETITIVE RE-PROCUREMENT
MAINTENANCE - 60% ADJUSTMENTS ELIMINATED	EMULATION APPROACH MAKES USE OF MODERN, AVAILABLE TECHNOLOGY
TRAINING REDUCED FOR O&M	
SYSTEM PERFORMANCE ENHANCED	

*$ 1 M PROGRAM - ONE YEAR

OLD DESIGN HAD TO BE FERRETED OUT PRIOR TO EMULATION

FIGURE 16

FIGURE 17

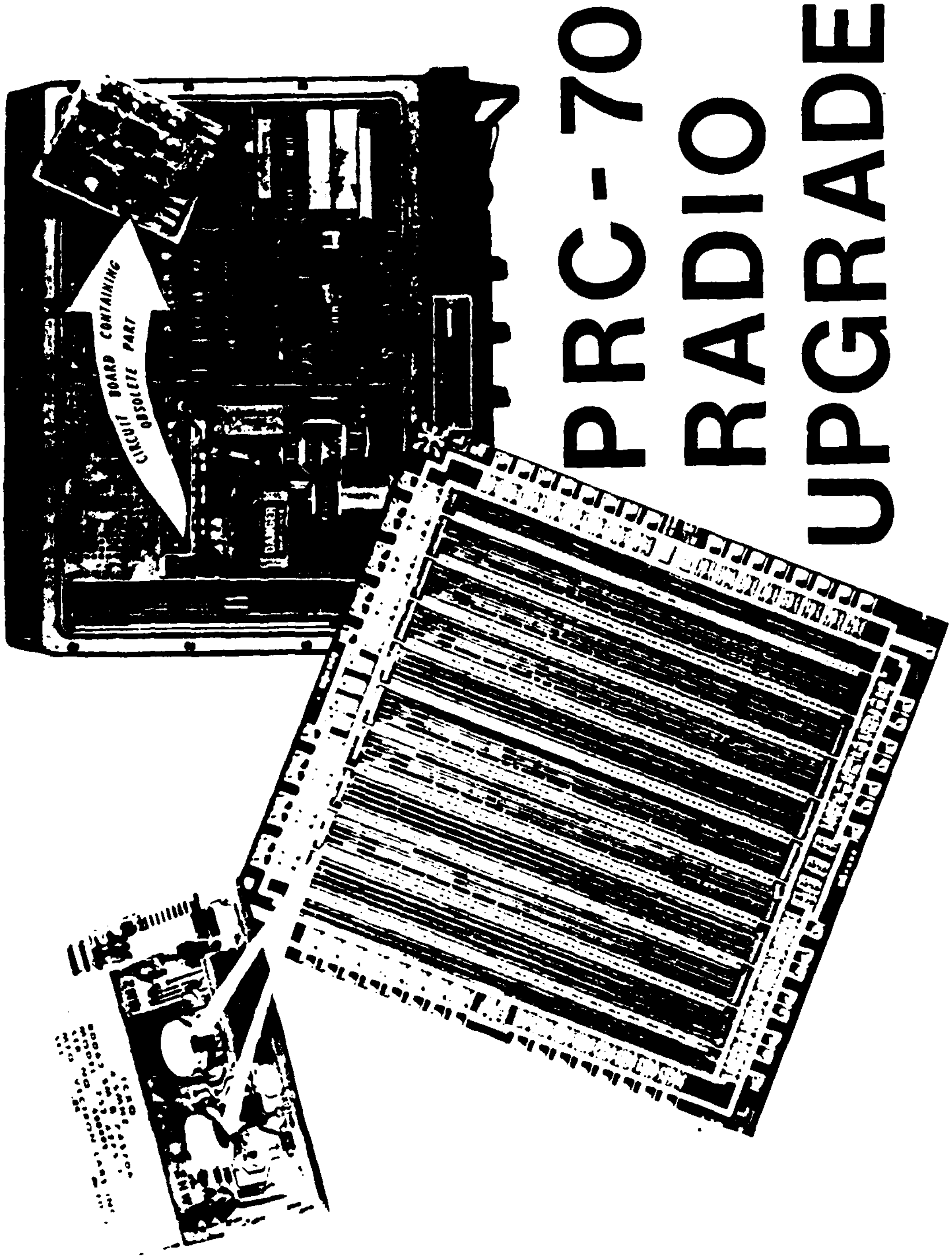

PRC-70
RADIO
UPGRADE
CIRCUIT BOARD CONTAINING
OBSOLETE PART

FIGURE 18

AUTOMATED DESIGN SYSTEM

1. RULES FOR DESIGN
   (accumulated knowledge of previous designers)
2. GLOBAL DATA BASE
   (library of previous designs)
   (library of algorithms)
3. AUTOMATED DESIGN SYNTHESIS
4. ASSESSMENT SYSTEM
   (design optimization)
5. HIERARCHICAL SIMULATION
   (design verification)
6. LIFE CYCLE COST MODEL

FIGURE 19

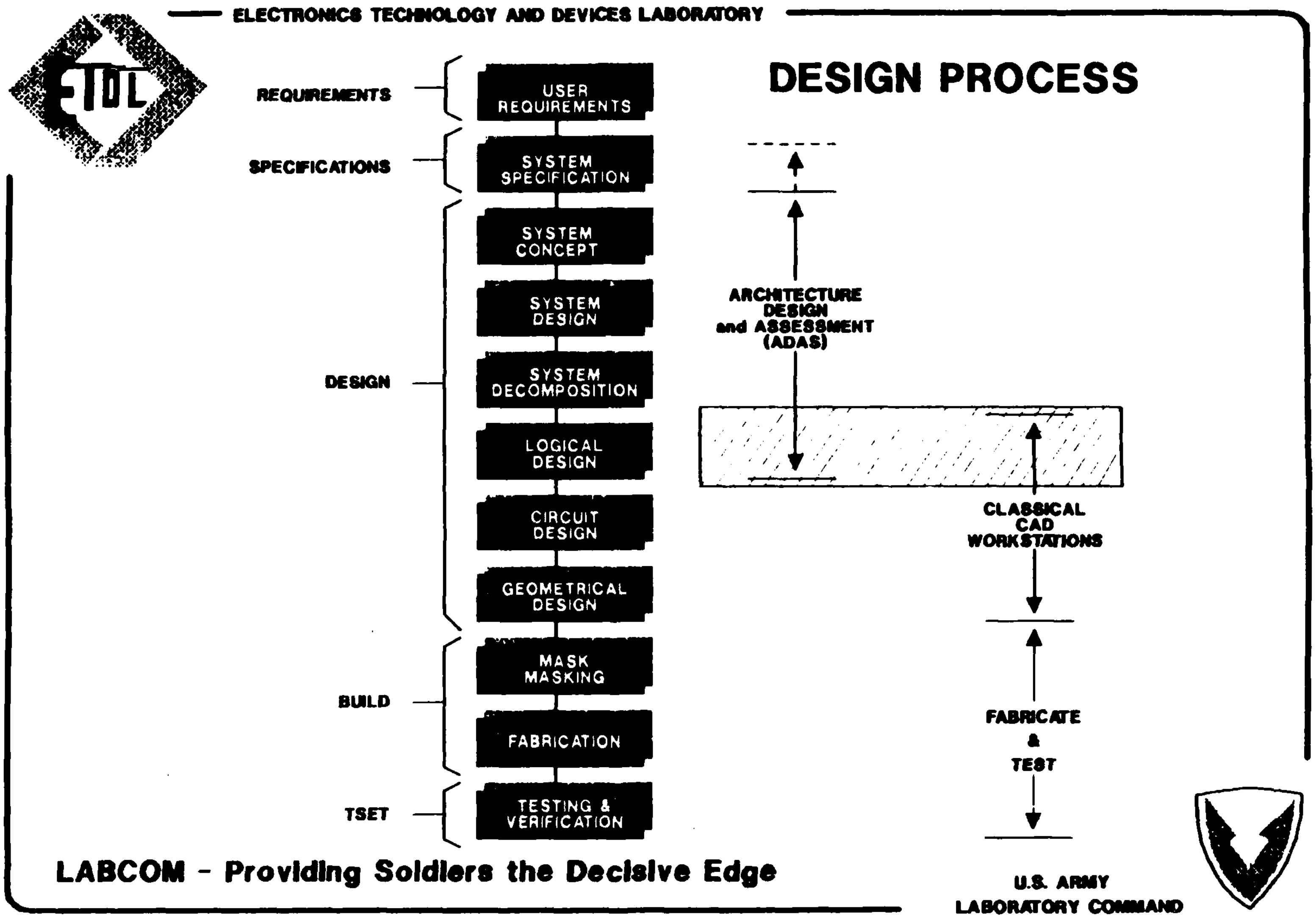

ELECTRONICS TECHNOLOGY AND DEVICES LABORATORY
ETDL
DESIGN PROCESS
REQUIREMENTS
SPECIFICATIONS
DESIGN
BUILD
TSET
USER REQUIREMENTS
SYSTEM SPECIFICATION
SYSTEM CONCEPT
SYSTEM DESIGN
SYSTEM DECOMPOSITION
LOGICAL DESIGN
CIRCUIT DESIGN
GEOMETRICAL DESIGN
MASK MASKING
FABRICATION
TESTING & VERIFICATION
ARCHITECTURE DESIGN and ASSESSMENT (ADAS)
CLASSICAL CAD WORKSTATIONS
FABRICATE & TEST
LABCOM - Providing Soldiers the Decisive Edge
U.S. ARMY LABORATORY COMMAND

FIGURE 20

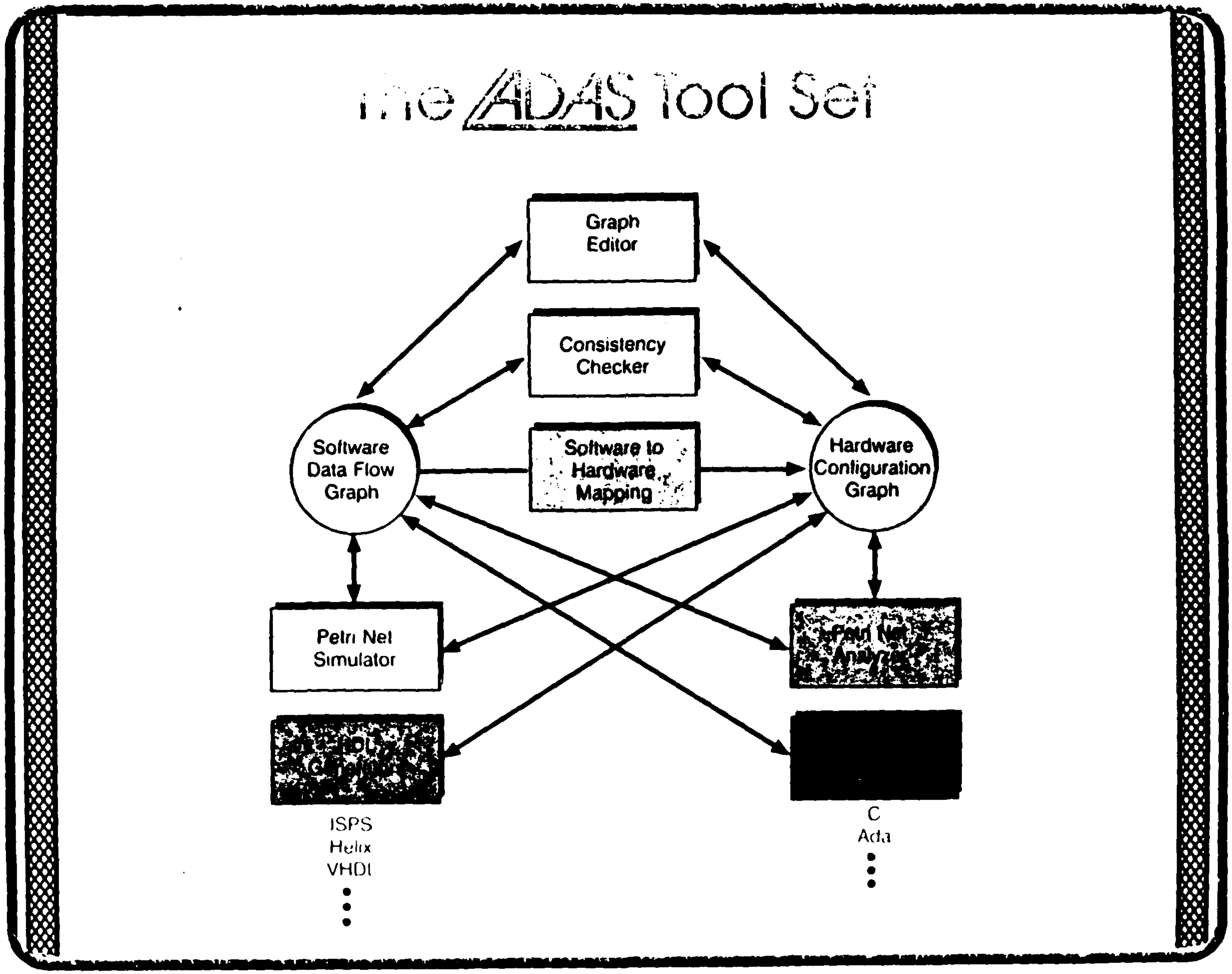

ADAS Tool Set
Graph Editor
Consistency Checker
Software to Hardware Mapping
Software Data Flow Graph
Hardware Configuration Graph
Petri Net Simulator
ISPS
Helix
VHDL
C
Ada

FIGURE 21

COMPUTER AIDED ENGINEERING FOR TESTABLE SYSTEMS

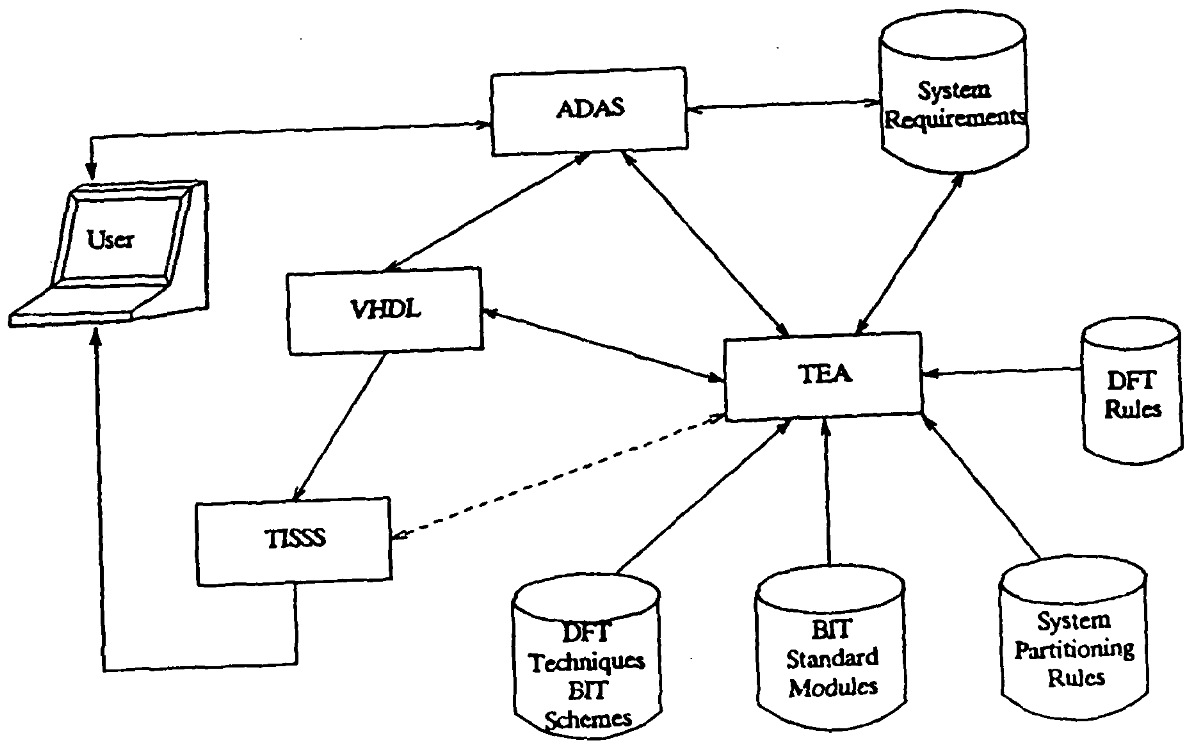


FIGURE 22

BENEFITS

1. PERMITS COMPETITIVE REPROCUREMENT OR UPGRADE AND SPARE PARTS PURCHASE.

2. UTILIZES PREVIOUS DESIGNS TO BUILD MORE COMPLEX SYSTEMS.

3. PROOFS DESIGN PRIOR TO FABRICATION.

4. "FUNCTIONAL DESCRIPTION" ALLOWS VENDOR FREEDOM TO INNOVATE.

FIGURE 23

ity rules (i.e., a rule base of guidelines that have been established by the military) built right into the system, the system will be capable of recommending built-in-test capability. It will aid in the partitioning of the system into testable units. It will be linked to another capability, that VHSIC is developing, called TISSS -- a Testing Independent Software Support System -- that has the capability to do some of the fault simulation in fault models. So, this is one of the efforts that we, in DoD, are working on to pull some of the problem solving up to the system level. VHSIC has done a lot of work in testability for chips, but if you don't use the right technique at the system level, it won't matter. All in all, the benefits (FIGURE 23) to us on these things that I've quickly dumped on you means that we need a methodology that allows us to rapidly generate or upgrade our hardware software, at any point in time, in affordable ways. We need to be able to save money by being able to use previous designs. We're hoping that by capturing designs in software we will be able to build a knowledge base into the computer aided design system that everyone can utilize ... both you and us.

I guess my closing remark is that we would like to work together to create a new way of doing business. It doesn't mean that you are all going to get a contract, because we don't have a lot of money. What it does mean, however, is that we need to work together to define our problems which need to be solved. We need to be able to find out together if we can develop the necessary CAD capabilities because there are billions of dollars of opportunity. There are people who will pick up on this ... this idea of using the VHDL. I believe there are billions of dollars of possibilities because we have a big company, and despite current fiscal problems, there is a lot of money, and there are a lot of opportunities. We believe we can interpret some of the new technologies that are coming along and can interpret some of the new ideas in CAD methodology. To interpret those new ideas to military people you have to demonstrate that the ideas really work.

WELCH: One of the things that I saw in the picture was an eight foot stack of documentation. It strikes me that in your hardware design languages, that's going to fall through the cracks if you don't keep in mind, in designing these languages, that somehow or other you've got to translate this information into documents that for 10 or 15 or 20 years, people out in the field are going to be using.

REITMEYER: There are other programs in DoD ... there's a program called "Computer-Aided Logistic Support," where people are trying to capture current documentation on tapes; the paper is not going to go away. I guess the point I was trying to emphasize is that there is work going on and the point I didn't want to miss is that a lot of information that has slipped through our fingers all these years -- namely, the design of systems. The point of this message is that we're paying for it and we should be getting it, and then anybody can make use of the design information for the future.

REED: I have a comment to make on that, too. I don't know whether you know it or not, but historically, hardware description languages were originally invented not for a computer but for the human being to use, to make it easy for the human being to design a shorthand notation. One of the first hardware description languages used digital logic, and it was digital logic which

was used in the development of many of the early computers. Then they made an extension of that, it was called the register transfer language, and I developed an extension of that. We use that to describe very accurately, independently of the machine hardware, the actual logical and architectural structure of the machine, prior to its being built. In fact, one machine that was built as long ago in 1957 or '58, which had the capacity of a modern small calculator, was designed completely in that manner and we actually simulated that language -- it was programmed then the register transfer language -- and it was programmed into the only existing machines of that time which accepted that program, which was called the TXO at the MIT Lincoln Laboratory. That machine that we designed was actually completely simulated in this hardware description language. And my comment on this ... what you said, languages you are inventing are designed so that human beings can communicate with the machine. In other words, human beings have to be able to communicate with one another -- in their designs and in their system design -- through this, by means of a shorthand hardware description language. To design a system properly you have to use that consistently. You just can't have one person use one hardware description language and another person use another one; you have to use a language which all people understand, including the machine that you put this hardware description language into. Then, once you do that, you can start designing systems and adding to them, modifying them, and coming up with a system which ultimately could be the most efficient system. Until you do that, you'll never succeed.

REITMEYER: Right. I think we agree on a couple of things. One of the things is that we can't afford to do it the way we're doing it now; we're just running out of money. We're running out of the bank. There are other efforts going on right now to make it easier, for example to develop code. There are workstation capabilities that are being funded to allow a designer to work at a symbolic level, where the system automatically generates the code.

Secondly, the IEEE right now is working on balloting -- maybe some of you are involved -- they're balloting right now to make an IEEE standard of the VHDL capability. Hopefully, that is a step in the right direction, but ultimately we've talked to educators and some universities about it. We have to start teaching throughout the educational community what this is all about. In fact, we've spoken to some of our key educators at West Point and we've said to those folks that you've got to be able to educate our future generals and colonels that there are new methodologies and ways of doing business in the military. We can't do it by just developing one or two tools. We think it's also an educational problem.

REED: Just one further comment on that. It still exists that you have to have a language that you can utilize as a human being on a piece of paper, where you utilize your own thought process, and then you communicate that to the machine, then the machine just uses external memory to help record what you thought about. In other words, using computer graphics today, people actually interact with the machine using a hardware description language. One of the things that's happening in modern times is that people have actually forgotten how to use logic design. They actually design employing methods that were used prior to the time that we used logical design. The

modern computer designer, the modern analog designer, all use what I would call a see-to-the-past design technique. That is not transferable. That technique is rarely, if ever, transferable to what I would call a hardware description language. You have to teach that hardware description language or invent one that somebody or all of us could utilize in order to accomplish what you want to do.

REITMEYER: I'd like to discuss that further. It's a good point. Any other questions?

CHOMA: As I mentioned before Charles McGuire of EESof is not going to be able to be with us, but Carl Ryan, our last speaker of the afternoon from Motorola, has been kind enough to offer a demonstration that I guess is somewhat comparable to the EESof demonstration -- sorry, not comparable, but nevertheless a demonstration. Carl Ryan from Motorola ....

RYAN: *Abstract*. The personal computer can be used as an effective tool in all aspects of communications system hardware development. The combination of low cost, wide availability and impressive processing power have proven to be effective in computer aided design applications that extend from the preliminary analysis to the prototype performance evaluation.

Performance capabilities of the PC have increased to the point where detailed communications systems simulations programs like "MODEM" process data at nearly 50 bits per second, a five fold increase in speed in three years.

Introduction. Within the past ten to fifteen years the typical communications scenario has grown from a fairly straightforward link involving a white gaussian noise channel and low complexity hardware to one which may involve a tightly bandlimited and nonlinear channel, power and/or bandwidth efficient modulation, error correcting coding, and receivers using adaptive signal processing techniques. In attempting to analyze or design such a complex system, one immediately faces the need for analysis tools which can rapidly evaluate and display performance characteristics, and facilitate the optimization of system parameters. Computer simulation is evolving into a powerful and flexible tool for realizing the above goals. The advent of the personal computer has greatly improved the accessibility and justification for developing simulation capabilities in any organization which has to deal with complex communication systems.

Several critical issues must be considered in the selection of a computer simulation tool. The most dominant parameters are:

(1) Computer accessibility
(2) Computer cost
(3) Hardware emulation capability
(4) Simulation accuracy
(5) Processing speed
(6) Capability for user written programs

The above issues are of major consideration for the simulation tools developed to support a variety of very wide band communications equipment programs at the Motorola Government Electronics Group. This development includes the design tools necessary to carry the task from the overall system to the printed circuit board design. The prototype assembly operation will be completed this year (1987), and in 1988 the prototype test functions will be completed.

Some of the issues to be addressed in this paper will be the signal processing capability of the PC as well as the utiliza-

tion of this equipment. A brief description of the communications simulation program "MODEM" will also be provided.

Computer Use by EE's. The use of computers by EE's can be broken down into five major categories:

(1) Bookkeeping and report preparation
(2) Drafting
(3) Circuit analysis and design
(4) Communication system analysis and design
(5) Software development

Discussion with engineers involved in the Communications System Development cycle indicates that the major amount of computer utilization is bookkeeping, report preparation and drafting. Very little time is spent in communication system analysis. A survey presented in the May, 1987 issue of the *IEEE Spectrum* (1) concerning CAD software availability provides additional insight into the problem of CAD for communication systems. The survey presented availability of CAD packages, none of which provide capability to perform Communications System Simulations.

Some of the reasons that the computer is not used more for the communications system design include:

(1) Program unavailability
(2) Computer unavailability
(3) Computer not needed

These general excuses for low utilization of the computer represents a sad State-of-Affairs in the State-of-Art of computer aided design of complex systems.

This problem needs to be addressed in order to find better methods to increase the CAD utilization by engineers. Perhaps the primary problem is the special capital equipment category used for computers. This category makes it unreachable by many R&D engineers. The computer is not treated as another piece of test equipment that is required to evaluate the communication system. Typically, in an R&D environment an average engineer will have direct control of more than $20,000 worth of commercial test equipment to perform his task; however, he must share a $1,000 computer with several engineers.

The PC Power. The IBM PC, the various clones and hardware add on's have provided an impressive array of technology for performing the various CAD tasks and in a cost effective manner. The clock speed available has increased from the original 4.77 MHz to 12 MHz, and the CPU processor upgrades from the 8088 to the 80286 and 80386 provided impressive increase in processing speed (2). These advances along with the availability of a variety of accelerator boards represent the hardware necessary to perform most CAD tasks.

The most powerful computing engines for personal computers appear, however, to be the 32-bit single board computers. These are available from several manufacturers and contain up to 16 megabytes of RAM and have clock speeds exceeding 20 MHz [3]. They provide a powerful simulation environment. These devices can be used in a single expansion slot of an IBM PC, XT, or AT and can therefore communicate with either an 8-bit or 16-bit bus structure.

Mike Fashano of Hughes Aircraft has verified performance roughly equivalent to a VAX 11/780 using the DSI 780+/4[2] single board computer in an IBM PC-XT running a FORTRAN program similar to a

[2] The DSI 780+/4 is a product of Definicon Systems, Inc., of West Lake Village, California.

large SYSTID simulation. Use of these devices enhances the simulation capabilities of a standard PC to that of a super minicomputer for many types of problems.

The Total Developmental Cycle. One of the deterrents to engineer usage of computers is the compatibility of the computer and the software available to carry a communication system from the initial system design through printed circuit board layout and ultimately to final assembly and test.

Effective computer use by engineers can be enhanced by using common CAD techniques throughout the development process.

The basis flow chart for the concept under development at Motorola GEG for wide band communications systems is illustrated in FIGURE 4.1. This concept is entirely based on the PC and makes use of custom generated programs that provide the engineer with the ability to compare and iterate designs and test results at various stages in the development process.

Critical in the final test process for the wide band systems is a computer positioned signal probe that does not require a direct ground. This probe has a bandwidth of DC to 4 GHz and allows for observing the various signals without disturbing the circuit in test. The precision probe placement based on the original computer generated assembly diagrams reduces probing errors and circuit board damage due to excessive mechanical stress encountered by hand signal probing. This test arrangement allows for direct comparison with computer generated signal waveforms and the measured signals.

The Modem Simulation Package. MODEM (4) is a fixed-topology simulation program written in PASCAL under the DOS operating system. Analysis of the modeling techniques used in MODEM indicate that accuracies on the order of 0.1 dB can easily be achieved. The program is completely menu driven and allows user developed models to be integrated into the simulation package. Typical processing speeds of 5 to 50 bits per second of simulation run time allow detailed system simulations to be accomplished with only a few minutes of computing time. The program requires an IBM PC or compatible with a math coprocessor, CGA graphics card, 256K of RAM and a printer. The source code is approximately 14,000 lines.

Program Capabilities. The MODEM simulation program consists of a series of simulation and support subroutines tied together by a short main program. Each of the analysis subroutines simulates one of the functional elements of the system illustrated in FIGURE 5.1.

The menu of available subroutines is illustrated in FIGURE 5.2. This menu allows the operator to evaluate a large variety of design approaches to a specific application including hardware design limitations. The results of the simulation can be displayed using the CRT monitor, printer or plotter. The dot matrix screen dump program and the pin plotter driver program are resident within MODEM so that no additional software is required in order to generate graphical outputs.

Typical processing speeds of the program are given in TABLE 5.1. Fast processing speeds are made possible by selective use of machine language code to complement the compiled PASCAL code. Error sources are identified in TABLE 5.2.

Program Structure and Execution. The main program (MODEM), as well as the graphics data management subroutines and one complex baseband signal set, are

resident in memory at all times. All simulation subroutines and data files for signals, filters, and all simulation results can be stored on disk if the user desires.

The simulation is based on the generation and processing of a complex baseband signal. The data for these signals is arranged as two arrays representing the I-channel and the Q-channel components of the baseband signal. Signals are stored at integer data to reduce memory requirements. FIGURE 5.3 shows the general program flow during execution of a simulation. The main program consists of a directory which is used to configure the simulated system by accessing a sequence of subroutines.

The subroutine structure provides its own menu of operations. Each time an operation is completed, control returns to the menu. From here the user can either stay in the subroutine for additional operations, or return to the main directory. Outputs for the simulation are generated by accessing routines which are memory resident. The ability to generate eye diagrams, envelope and excess phase diagrams at any point in the simulation is a significant aid in optimizing performance of the communication system under study.

In performing an actual simulation the user generates a code sequence (typically a PN sequence) of desired length using the code generator. Filters can also be defined at this time. The modulator subroutine is used to generate a complex baseband signal from the code sequence. Different systems are modeled by processing the modulated code sequence through a selected cascade of subroutines, any of which can be used more than once. Intermediate results can be stored on disk at the end of each subroutine. At any desired point, the signal can be run through the bit error detector. This model generates performance curves versus received E_b/N_0, the carrier phase reference, sample time or the decision threshold. The time and frequency domain subroutine can also be used at any point to view power spectral density, signal correlation properties or other signal properties. FIGURES 5.4 through 5.7 illustrate typical simulation outputs.

Parameter Normalization. All time and frequency variables have been normalized to the serial NRZ data rate. Thus, all parameters relating to sample time, filter bandwidths and center frequency must be selected on the basis of this data rate normalization.

The signal amplitude is normalized to unity and uses integer representations for memory storage with unity assigned the value of 10,000. This integer representation of the signals requires 75 percent less memory than would be required using the corresponding real numbers. This scale factor of 10,000 gives an 80 dB dynamic range so that the impact on system performance is negligible.

Modeling Techniques. All signal filtering, as well as nonlinear and time varying signal processing, is performed on the baseband signals in the time domain since this represents the actual hardware designs more closely and the simulation routines are easier to develop. The speed advantage of frequency domain processing of the linear filtering operation does not provide sufficient performance incentive over the time domain approach to justify its use.

Code Generation and Modulation. The initial step in the simulation is to generate the data signal. This signal is typically a maximal length PN sequence,

although manual entry of a specified data sequence is possible. Data symbols can be grouped in order to form M-ary waveforms such as QPSK and 64/QAM.

The serial data stream from the code generator is then used to modulate a carrier to yield the signal

$$c(t) = A(t) \cos [2\pi f_D t + \phi(t)] \quad (1)$$

where f_D is the carrier frequency and $\phi(t)$ is the phase. The power in $A(t)$ is normalized to unity for all modulation formats. The modulated signal, $c(t)$, is decomposed into direct (I-channel) and quadrature (Q-channel) signals for use throughout the remainder of the simulation. A large number of modulation formats are menu selectable including PSK, QPSK, OQPSK, 8PSK, 16PSK, MSK, 16QAM, and 64QAM.

Filter. The filter program generates filters in a two step process. First a set of amplitude and group delay samples are generated corresponding to the frequency characteristics of the desired filter. The complex impulse response is then calculated by performing a discrete Fourier transform on the amplitude and group delay samples. Filter response normalization is achieved by making the amplitude response of the filter unity at the desired reference frequency.

Linear and Non-Linear Signal Processing. Filtering operations are performed by numerically convolving the I/Q signal with the complex impulse response of the filter. The transmitter effects of bandpass nonlinearities are modeled in the transmitter subroutine. After the signal has been convolved with the transmitter filter, the envelope amplitude is calculated and the phase shift parameter is determined.

The envelope amplitude and the AM to PM conversion coefficient of the nonlinearity is used to rotate the signal constellation, which is then passed through a selected amplitude transfer function. This approximation provides a satisfactory model for many nonlinearities such as traveling wave amplifier.

Bit Error Rate. The bit error rate is calculated as a function of (1) E_b/N_0, (2) carrier phase, (3) sample time, and (4) decision threshold. The operator is prompted by the computer to select the desired parameter to vary, the range of the variable, the starting point, and the resolution. The noise bandwidth of the receiver filter is used as a reference for computing the bit error rate. The operator can select this parameter if desired. The remaining parameters are selected by responding to computer generated questions. The data computed from this program is displayed in graphical form or as a listing. FIGURE 5.8 illustrates the results obtained from this subroutine.

The bit error rate calculations are based on the evaluation of the complementary error function with the signals obtained from a histogram of data samples which correspond to the three operations that are performed in the subroutine. These are:

(1) Interpolation of the data points to correspond with the selected sample time.

(2) Construction of a histogram of the sample points.

(3) Computation of the resulting BER from an approximation to the complementary error function.

The interpolation routine uses a four point LaGrange method, which is sufficiently accurate providing at least 4 samples per symbol are used to simulate the data.

Error Correcting Coding. The impact of using error correcting coding is evaluated by using the relationship

$$P_{ec} = A\,(P_{eu})B \qquad (2)$$

in which P_{eu} is the channel symbol error probability without coding and P_{ec} is the system bit error probability with coding. The parameters A and B are constants which depend upon the particular code under study. Equation (2) is simply a consequence of the fact that the relationship between log P_{ec} and log P_{eu} is essentially linear for sufficiently small values of P_{ec} and log P_{eu}. The appropriate values of A and B must be determined separately for the code of interest. Once they are known, however, this procedure gives good results for error probabilities on the order of 10^{-3} or less and is extremely useful for establishing comparative performances between various coding strategies.

Euclidean Distance. The Euclidean distance routine provides the operator the ability to compare the Euclidean distance d_{ij} between any two signals $s_i(t)$ and $s_j(t)$, defined by

$$d_{ij}^2 = \int_{T_1}^{T_2} |s_i(t)-s_j(t)|^2 dt \qquad (3)$$

The operator selects the signals $s_i(t)$ and $s_j(t)$ as well as the integration interval, T_1 and T_2. The results are displayed in numerical form.

Power Spectral Density. This subroutine is designed to provide the operator the ability to evaluate the signal conditions at any point within the simulation. The power spectral density calculations for this subroutine are based on the Fourier transform of the autocorrelation of the signal. The first operation to perform is the computation of the auto and crosscorrelations of the inphase and quadrature signals.

The parameters of importance in this analysis are the duration of the significant correlation coefficients and the desired resolution of the computed power spectral density. The operator is prompted to select these parameters and should choose them on the basis of an estimate of the signals under investigation. The duration of the autocorrelation should be selected to be only slightly longer than the best estimate of the actual autocorrelation duration. This is important since the time required to perform the calculation is proportional to the autocorrelation duration. The frequency resolution and frequency range should be carefully selected for similar reasons. FIGURE 5.9 illustrates typical results.

User Written Subroutines. The simulation program has been configured to allow for user generated programs that represent analysis routines and models not included in the basic program. Typical user-defined programs include carrier and symbol synchronizers as well as alternate adaptive equalizer designs. All of the routines resident within the MODEM program and all of the signals generated within MODEM are available to the user-defined programs. As always, a valid mathematical model, and the appropriate source code to realize the model, are necessary for all routines supplied by the simulation user. Since MODEM uses PASCAL source code, a compatible PASCAL compiler is required in order to develop the object code for user supplied models. Typical user written routines require 50 to 500 additional lines of PASCAL source code.

Conclusion. Computer-aided design of communications systems can be achieved in a very cost-effective manner using the PC. The use of the PC for these types of CAD

Figure 4.1 Computer Aided Design, Assembly and Test Concept

	$2,000	$8,000	$60,000	$100,000
SYSTEM REQUIREMENTS	MODEM	CIBLOA	CAAR	TES Assembl
and subsystem	Analyze and design system and analysis	Circuit board layout	Assembly and Repair controlled	Test Ass remote signal r
	WAVEFORMS	WAVEFORMS		WAVEFORM
	POWER SPECTRAL DENSITY			PSD
	BER			BER
		MAKE LAYOUTS	PICK & PLACE	REDESIGN
		CUT RUBY	REPAIR	

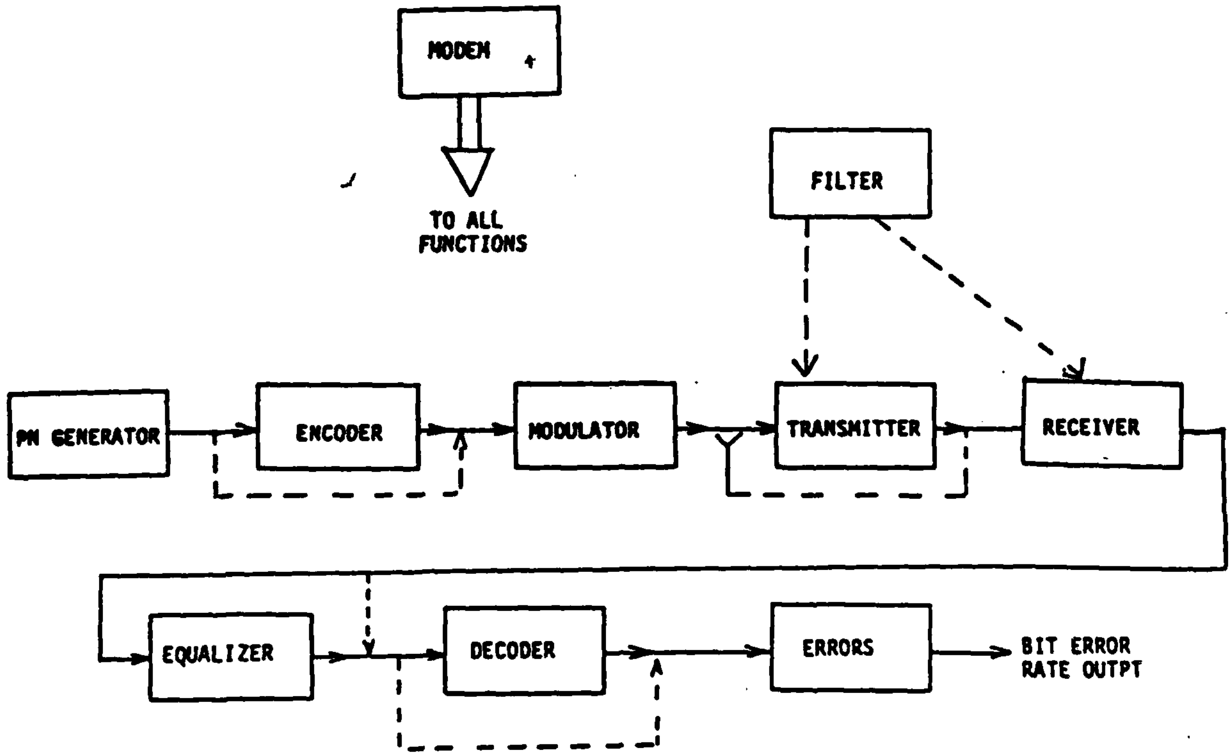


Figure 5.1 MODEM Simulation Functional Block Diagram.

```
---------------------------- MODEM  (c) by Carl Ryan --------------------------

THIS PROGRAM CONTAINS THE FOLLOWING FUNCTIONS:

      (1)    PN Generator
      (2)    Data Encoder
      (3)    Modulator
      (4)    Filter
      (5)    Transmitter
      (6)    Reciever
      (7)    Error Detection
      (8)    Time & Frequency Domain Plots
      (9)    Transversal/Recursive Adaptive Equalizer
     (10)    Signal Combiner
     (11)    Euclidian Distance Bounds
     (12)    Cross Correlations
     (13)    Reconfigure Initialization for "MODEM"

TYPE THE NUMBER OF YOUR CHOICE (1-13) :

    F1      F2      F3      F4      F5     F6      F7     F8     F9      F10
   Forwd   Bckwd  XitSub  XitPro   LOAD  DiskDr  PrtPg  DmpGr  PlotGr   Stop
```

Figure 5.2 MODEM Main Menu and Control Screen.

TABLE 5.1 Processing times for MODEM Program

512 Data Bits MSK signal filtered with
a 5 pole Butterworth filter
4 samples per Data Bit

	PROCESSING TIME IN SECONDS				
	COMPUTER TYPE				
PARAMETER	IBM XT 4.77 MHz	TURBO XT 8 MHz	IBM AT 6MHz	IBM AT 8 MHz	COMPAQ II 16 MHz
Data Generation	1	1	1	1	1
Modulation	8	5	7	4	3
Filter Construct	8	5	6	4	3
Signal Filter	15	9	20	10	9
Signal Storage (Floppy Disk)	22	15	8	9	4
Signal Retrieval (Floppy Disk)	10	7	6	4	4
Power Spectral Density	28	17	37	17	17
5 Section Adaptive Equalizer	71	44	44	28	18
BER vs					
E_b/N_o	3	2	2	1	1
Bias	8	5	5	3	2
Phase	8	4	5	3	3
Timing	14	9	10	6	4
Total Time	196	123	151	90	71

TABLE 5.2 SOURCES OF SIMULATION ERROR IN MODEM PROGRAM

Time Resolution	Ailising	.05 dB
	Interpolation	.05 dB
Amplitude Resolution		.001 dB
Finite Impulse Response Filters		.05 dB
ERFC Approximation		.01 dB
Noise Bandwidth Calculation		.05 dB
Signal Amplitude Calculation		.01 dB

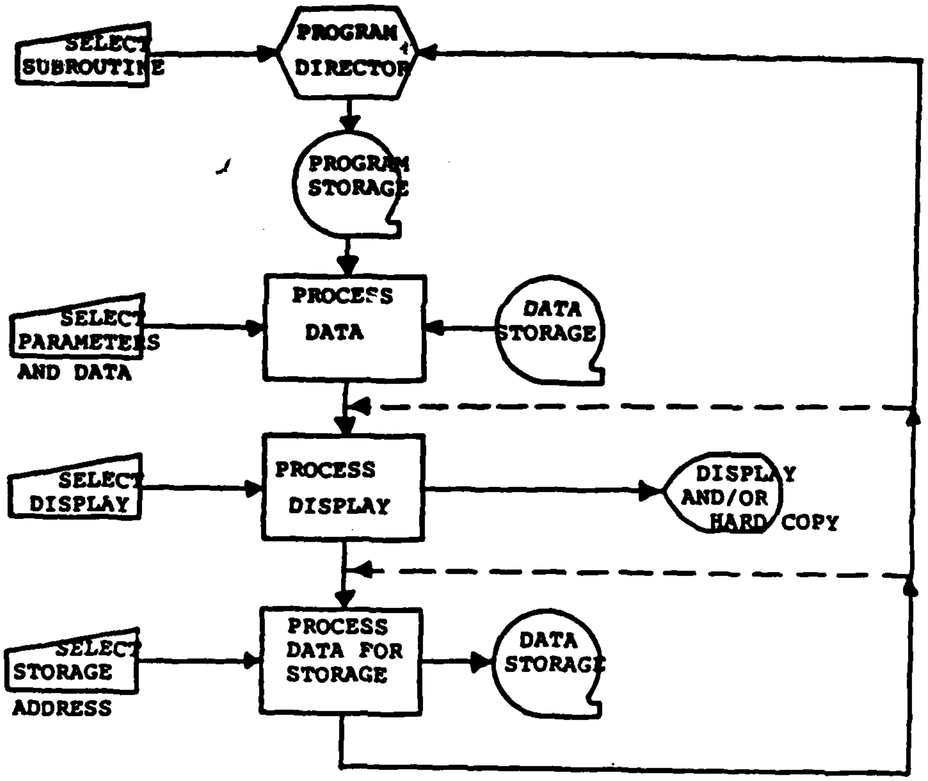


Figure 5.3 Program Flow During Execution of MODEM.

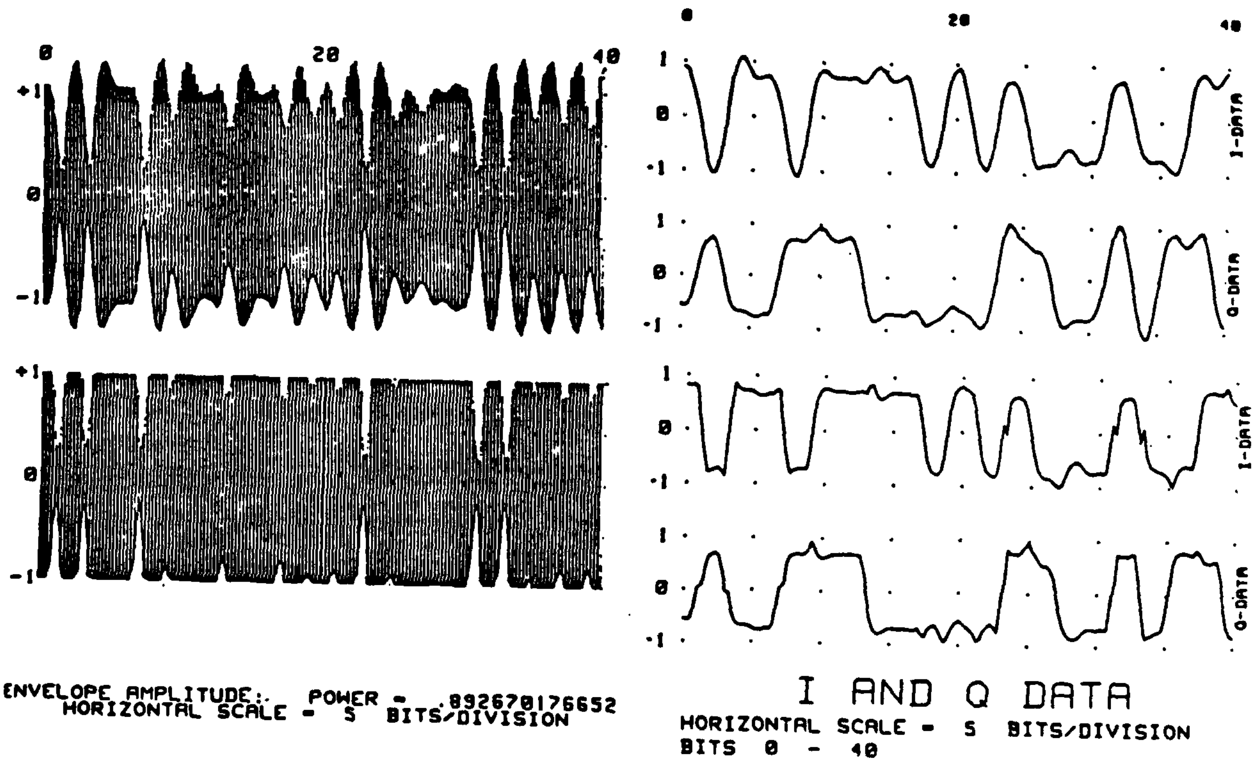


Figure 5.4 QPSK Signal Before and After Power Amplifier.

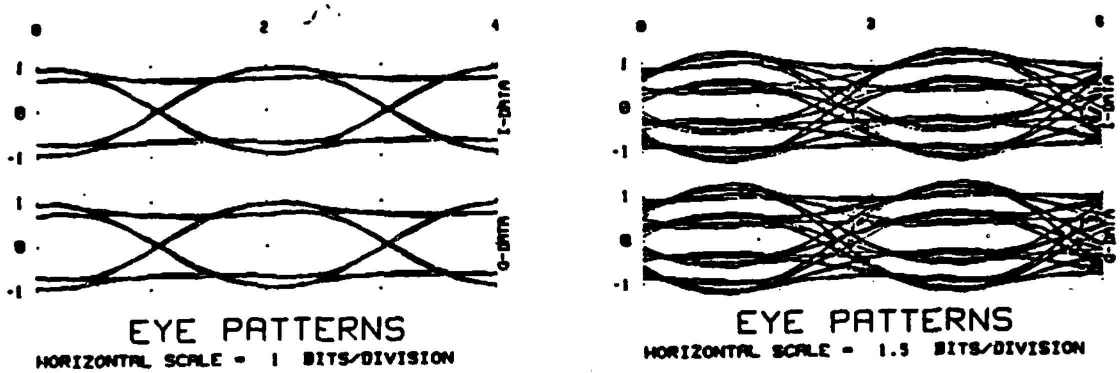


Figure 5.5 Eye Patterns for QPSK and 8PSK.

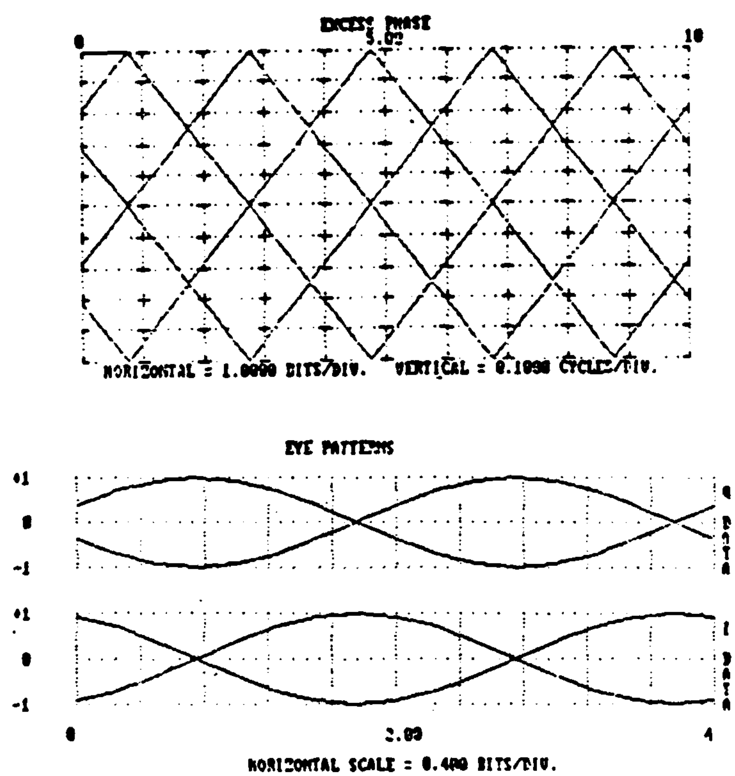


Figure 5.6 Phase Trellis and Eye Pattern for MSK.

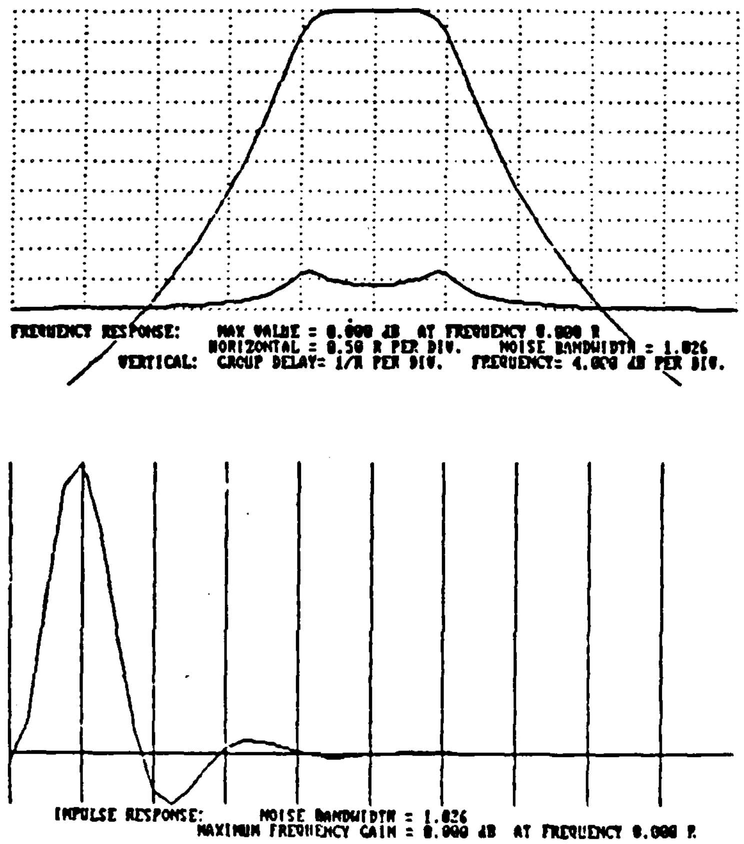
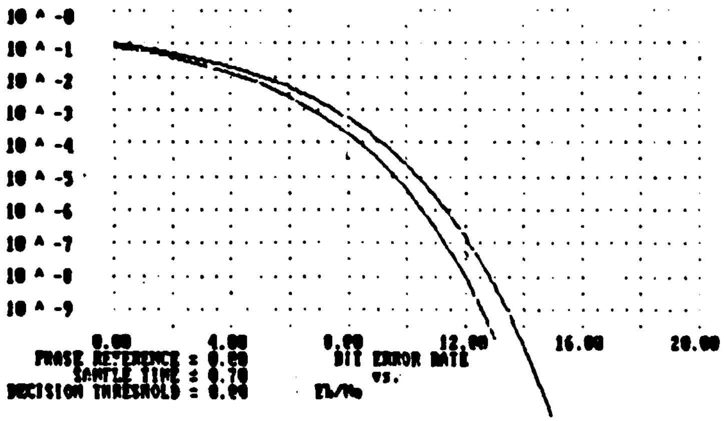
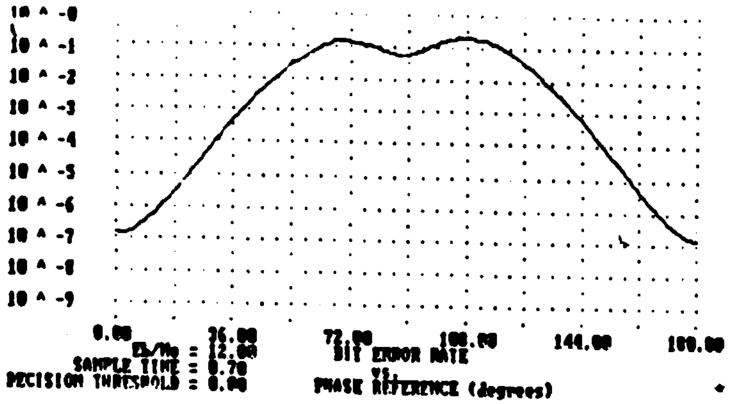


Figure 5.7 Frequency Response and impulse Response of Filter.

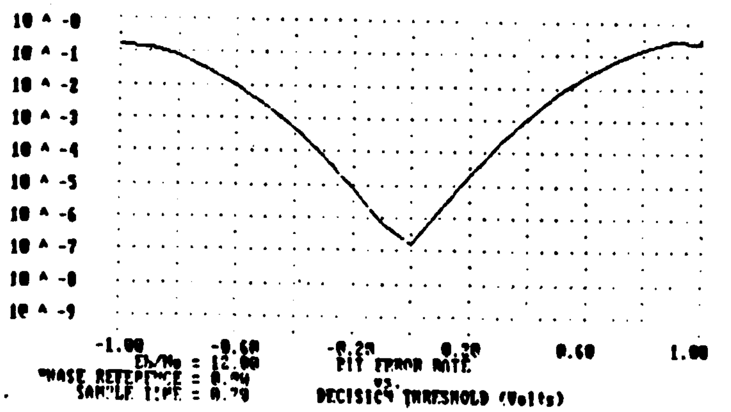


(a) BER as a Function of E_b/N_o.



(b) BER as a Function of Phase Reference.

(c) BER as a Function of Sample Time



(d) BER as a Function of Decision Threshold.

Figure 5.8 BER (Bit Error Rate) Parameter Options.

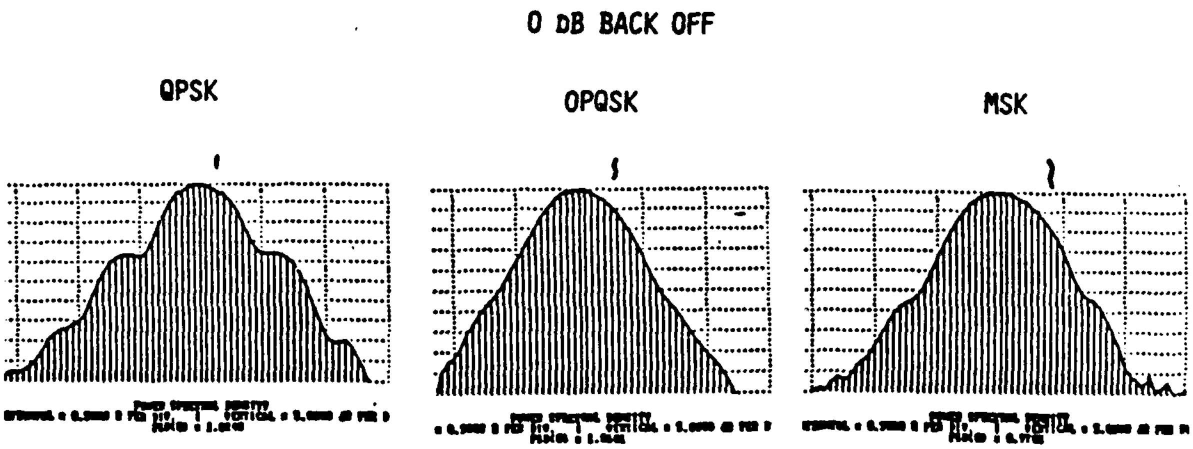
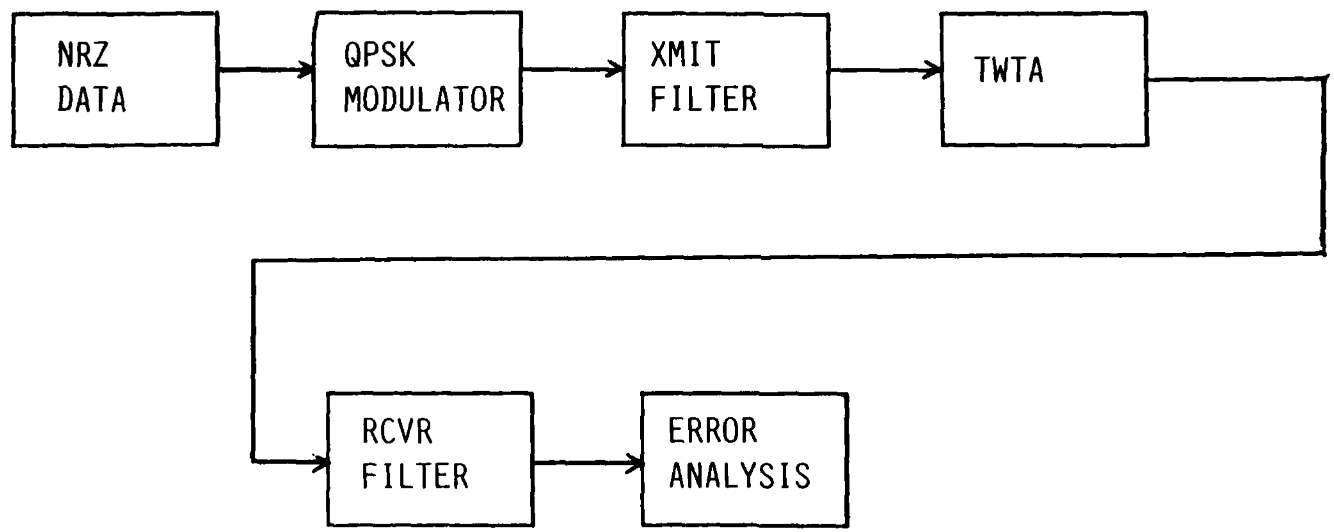


Figure 5.9 Power Spectral Density After Filter and TWTA.

APPENDIX A

Demonstration of Modem Program

The following set of charts represents a typical set of menu screens and data obtained from the Modem Program when evaluating the performance of the communication system illustrated in Figure A1. The Batch Mode of operation of the Modem Program allows one to easily generate and document a set of data as illustrated in this example.


NRZ DATA
QPSK MODULATOR
XMIT FILTER
TWTA
RCVR FILTER
ERROR ANALYSIS

Appendix A1

```
                    MODEM 4.06 - Batch Mode Version



                (1) Batch Mode
                (2) Rebatch Input File
                (3) Run Normal Modem




Select Option..: 3
```

requirements can be improved by increasing the general availability of the computer and providing a common CAD concept throughout the development cycle.

Simulation packages like "MODEM" and "CIBLOA" provide the initial portions of this common CAD concept.

References.

(1) Kate, L, "Automating Electronic Design," *IEEE Spectrum*, May 1987, pp. 55-58.

(2) Davis, S.R., "Clocking the 80287 at High Speed," *PC Magazine*, April 14, 1987, pp. 188-203.

(3) Davis, S.R., "Changing the Rules of the Game," *PC Magazine*, April 14, 1987, pp. 208-218.

(4) Ryan, C.R. and D. Jordan, "Communication System Analysis Program for the IBM PC," *IEEE Phoenix Conf. on Computer and Communications*, March 1986.

Appendix A: Demonstration of Modem Program. The following set of charts represents a typical set of menu screens and data obtained from the Modem Program when evaluating the performance of the communication system illustrated in FIGURE A1. The Batch Mode of operation of the Modem Program allows one to easily generate and document a set of data as illustrated in this example.

REY: I really have two questions. First of all, how do you add models? Is it hierarchical or do you have to write Fortran code for something like if you wanted, let's say, a 16-ary PSK or some strange signaling format?

RYAN: It depends on whether you have the original software and the original source code or not. If you have the source code then it would be no more than two or three lines of code and then you recompile the code. On the other hand, you can access our current database which through the disk storage stores the data on the disk. If you want to generate your own data through whatever mathematics you want to, you can generate that, store it on the disk in the same format that we sort of did. Then of course the MODEM program will pick up that data and process it. Take what bit error rate is, what power spectral density, and whatever else you want to do. So that allows you to write your own programs, your own user programs in whatever language you choose. All you need to do is be able to store it on disk so this program can pick it out and also you access it in the same way.

REY: Second question .... Can we buy this and how much is it?

RYAN: We can't buy it because we're not in that business, but I do give them away. I have given away ....

REY: I'll take a copy!

RYAN: ... to a couple of universities. I don't know whether I can give away to TRW or not, but [laughter] ....

REY: We'll sell you BOSS! [laughter]

RYAN: I'll tell you what, I'll make you a trade. You send me BOSS and I'll send you this.

REY: No, we won't make that trade ....

RYAN: We don't, we're not into that business. But if you are interested, let me know, send me a letter, but for universities we would probably give them that.

PURSLEY: Yes, I have a question. I'm trying to understand what is exactly being simulated and what's being calculated. Are you going all the way to the

receiver and cycling through all your bit patterns, everything like that, getting the sample values into the calculation of the error function?

RYAN: That's correct. Yes, what we do in doing the error calculation is we look and take a sample at each symbol time, find out what voltage that is, store that in a histogram, which of course is a histogram of the sample time, then you simply take the error function of the point in that histogram and then you have very good bit error rate.

PURSLEY: Now, this wouldn't allow you to handle nonlinearities after the noise is added?

RYAN: That's correct. And that's why you wouldn't have to go to a Monte Carlo type of simulation.

MOHANTY: Of the capabilities you have, you said that you have the capability to do hardware emulation. Do you have a computer that ...?

RYAN: Yes, what we do in order to handle nonlinearities, we have a Taylor series expansion. I think it uses 4 points for the amplitude and 3 for the phase. You have to figure out those points to put in there and then calculate it to be accurate enough.

MOHANTY: You said that you have from the major data, for example, you have the values from the amplitude spectrum and phase spectrum. Do you think you can identify the poles and zeroes and the number of poles and the number of zeroes?

RYAN: We've never tried that. The way the filter routine works is that we have of course classical filters, and then we have the major data. You can cascade the filters in whatever way you want to, and so all the cascading operations are done in the frequency domain, it multiplies the amplitude and adds the phase or group delay and then after you have cascaded all the filters you want, it simply takes Fourier transform and that's the impulse response, which is used in the rest of the program.

MOHANTY: But this, if you cannot identify from any major channel the number of zeroes and poles ...?

RYAN: I guess I'm not -- I don't care about it.

MOHANTY: Another thing I'd like to ask you, do you have any thing for coding, error correcting coding and decoding?

RYAN: Yes, we have a routine that works quite effectively in that.

MOHANTY: What kind of codes do you have?

RYAN: Oh, it doesn't, it's not dependent on the code. The way we do that in the error correcting, is somebody may tell you cheat, but we don't, it's very accurate results. What you realize is that most error correcting coding, particularly in hard decision error correcting code has a curve of bit error rate in versus bit error rate out.

MOHANTY: So you don't have any soft decision, Viterbi decoding?

RYAN: No, we don't have that. But once you have a curve of bit error rate in versus bit error rate out it turns out that over the range of like 10^{-3} on down to very low bit error rate, that is a straight line on a log-log plot, so what you really need is 2 terms. You need a coefficient multiplier and an exponent, and from that you calculate the bit error rate.

MOHANTY: How about fractional equalizer?

RYAN: Fractionally, if you can incorporate whatever fractions you want, it is softwared.

MOHANTY: And anything on quantization?

RYAN: Quantization resolution, how do you mean?

MOHANTY: I mean sampling and quantizing and taking the values when you digitize from the RF signals down to the band analog.

RYAN: We don't worry about that level.

MOHANTY: You take only baseband signals?

RYAN: Our signals are, they are really, everything is done from the RF domain or equivalent treated zero as the frequency. And everything is measured ±0. So it is all treated as an RF signal. Then you can look at the I and Q waveform. Any other questions?

CHETHIK: Do you have any provision for differentially coherent detection?

RYAN: The software that I have here is entirely for coherent detection. We have fiddled around a little bit with FM and differential coherent. And those are the kinds of modules that are too sloppy right now to get incorporated in. This is the way the program has grown. If somebody wants an FM detector or a differential coherent detector, we'll write one, we'll build one and decide if it has universal value in its implementation.

CHETHIK: I had some earlier sentiments and am very interested indeed in procuring.

RYAN: Any other questions? I had better give the demonstration now. Let me just preface the demonstration with by telling you that this is an entirely menu-driven program. What I have done on this demonstration is that I have already entered all of the questions in the menu program and those answers are stored on disk. So what this really means is that we are running in a batch mode now. What we are doing now is, we are processing about 300 data bits of QPSK signals. We will be filtering it. We will be going through an amplifier, we will be going through an adaptive equalizer, and we will be looking at bit error rate. I will try and narrate some of the things as we go along. It takes about 5 minutes. So I think if everything runs, we'll just start ... it should be booting up. Because of the fact that these questions are all answered, we probably won't be able to see the menu completely, and we won't be able to read them. Processing the data, this isn't information that is stored. Of course, this is a slow computer, we are not really advertising speed. This is an IBM PC ....

MOHANTY: When you determine these complex weights, do you have any algorithm to determine their weights, any particular algorithm you use?

RYAN: Oh, the equalizer is completely adaptive. The algorithm I'm using in there is a mean square error algorithm.

MOHANTY: Yes, I know, but that's the criterion. But what algorithm to determine the weights? I mean, is there any Wiener-Hopf equations or any particular algorithm you use?

RYAN: No, no particular. I just look at each path and I calculate it.

MOHANTY: Do you have any convergence factor when it will convert or anything? It converges everything all right?

RYAN: Yes.

MOHANTY: You don't have to worry about in timing or any adjustment of width?

RYAN: No, it does fine. As long as you -- that's when you want to look at the

```
                              MODEM 4.06

Configured for        :  IBM AT WITH A GRAPHICS MONITOR

Default Drive         = "Q"
  ault FileName       = ""

EGA Graphics will be used.

To access MEDIUM plots from the plotter use  [F9].
To access SMALL plots from the plotter use  [Shift] [F9].
To access LARGE plots from the plotter use  [Alt] [F9].

NO Automatic defaults will be used.

To re-configure MODEM, press [F2].
If configuration is correct, simply press [F1].


    F1     F2     F3     F4     F5     F6     F7     F8     F9     F10
  Forwd  Bckwd  XitSub XitPr
```

```
* * * * * * * * * * * * * * *
*                           *
*                           *
*   MODEM ANALYSIS  4.06    *
*   (Preliminary Release)   *
*                           *
*  (C) MOTOROLA INC. 1987   *
*           BY              *
*        CARL RYAN          *
*                           *
*                           *
* * * * * * * * * * * * * * *
```

```
  F1      F2      F3      F4      F5     F6      F7     F8     F9     F10
Forwd   Bckwd   XitSub  XitPro   LOAD  DiskDr  PrtPg  DmpGr  PlotGr  Stop
```

PROPRIETARY NOTICE

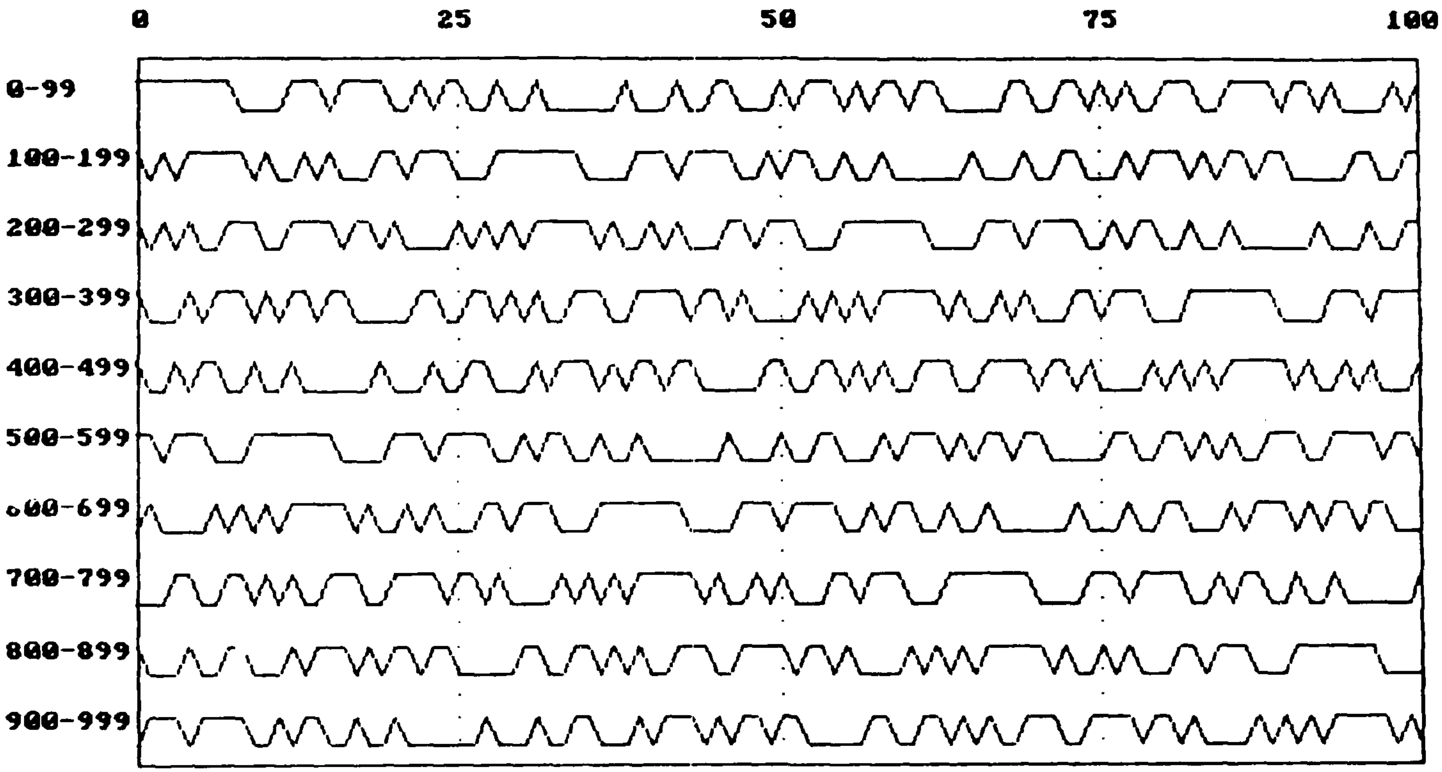
THIS DOCUMENT CONTAINS INFORMATION PROPRIETARY TO MOTOROLA AND IS SUBMITTED SOLELY FOR YOUR EVALUATION. THE INFORMATION HEREIN SHALL NOT BE DUPLICATED, USED, OR DISCLOSED FOR DESIGN, PROCUREMENT, MANUFACTURING, OR ANY OTHER PURPOSE, IN WHOLE OR IN PART, WITHOUT MOTOROLA'S WRITTEN CONSENT.

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Forwd	Bckwd	XitSub	XitPro	LOAD	DiskDr	PrtPg	DmpGr	PlotGr	Stop

```
------------------------ MODEM  (c) 1987  by Carl Ryan ----------------------

MODEM contains the following functions:

      (1)    PN Generator
      (2)    Encoder
      (3)    Modulator
      (4)    Filter
      (5)    Transmitter
      (6)    Receiver
      (7)    Decoder/Demodulator/Monte Carlo
      (8)    Error Detection
      (9)    Time & Frequency Domain Plots
     (10)    Transversal/Recursive Adaptive Equalizer
     (11)    Signal Combiner
     (12)    Euclidian Distance Bounds
     (13)    Discriminator
     (14)    FSK Receiver
     (15)    Reconfigure Initialization for "MODEM"
     (16)    Utilities
Enter the number of your choice (1-16) : 1

      F1     F2     F3     F4     F5     F6     F7     F8     F9     F10
     Forwd  Bckwd  XitSub XitPro  LOAD  DiskDr PrtPg  DmpGr  PlotGr  Stop
```

```
1/28/88  8:44 am

----------------------------------------------------------------------------
PN GENERATOR  (c) 1986  BY CARL RYAN   :
(1) PN GENERATOR (2) MANUAL ENTRY = 1  :
                                       :
NUMBER OF STAGES (3-33) = 7            :   MAXIMUM LENGTH CODES:
     XOR TAP LOCATION  = 4             :
                                       :   STAGE      XOR LOCATION
PREAMBLE ? (1=YES 2=NO) = 2            :     3           1,2
 PLOT ON SCREEN  (Y/N)  = N            :     4           1,3
MODIFY DATA (Y/N)       = N            :     5           2,3
Choose functions below.Exit when ready:      6           1,5
                                       :     7           1,3,4,6
                                       :     8           NO SINGLE TAP
                                       :     9           4,5
                                       :    10           3,7
                                       :    11           2,9
                                       :
                                       :
                                       :
                                       :
                                       :
                                       :
                                       :
----------------------------------------------------------------------------
```

1/28/88 8:45 am



7 STAGE GENERATOR, TAP 4 IS FED BACK.

F1
Forwd

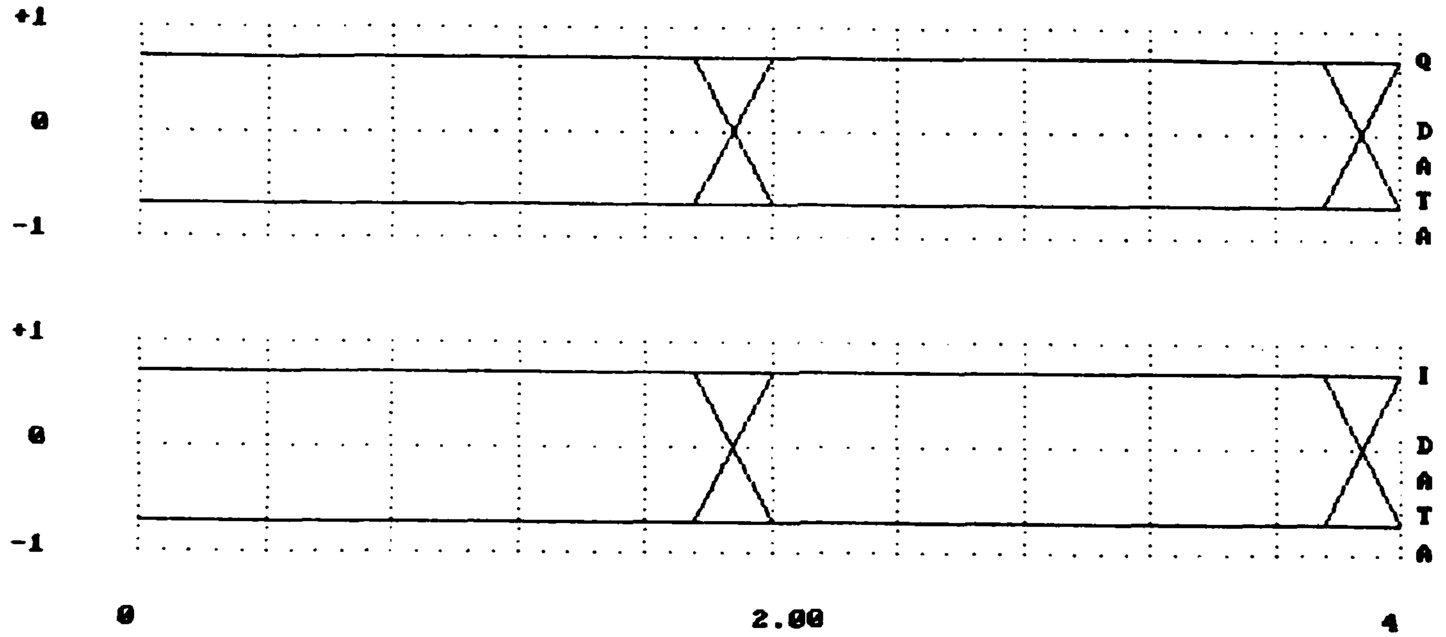
```
1/28/88  8:51 am

--------------------------------------------------------------------------------
------------------------ MODEM  (c) 1987  by Carl Ryan ----------------------

   MODEM contains the following functions:

         (1)    PN Generator
         (2)    Encoder
         (3)    Modulator
         (4)    Filter
         (5)    Transmitter
         (6)    Receiver
         (7)    Decoder/Demodulator/Monte Carlo
         (8)    Error Detection
         (9)    Time & Frequency Domain Plots
        (10)    Transversal/Recursive Adaptive Equalizer
        (11)    Signal Combiner
        (12)    Euclidian Distance Bounds
        (13)    Discriminator
        (14)    FSK Receiver
        (15)    Reconfigure Initialization for "MODEM"
        (16)    Utilities
   Enter the number of your choice (1-16) : 3

--------------------------------------------------------------------------------
```

1/28/88 8:52 am

```
--------------------------------------------------------------------------------
-- MODULATOR -- (c) 1985  BY CARL RYAN| OPTIONS :
Use PN Code in memory?  1)Yes 2)No = 1| 1) PLOT EXCESS PHASE TRELLIS
            MODULATION TYPES          | 2) PLOT EYE DIAGRAMS
 1) BPSK   2) QPSK  3) OQPSK  4) 8PSK | 3) PLOT "I" AND "Q" DATA STREAMS
 5) 16PSK  6) MSK    7) SMSK   8) KPSK | 4) PLOT SIGNAL CONSTELLATION
 9) SDBLP 10) CPM  11) 16QAM 12) 64QAM| 5) MODIFY FOR TDMA SIGNAL
13) BMSK  14) DPSK 15) TBA   16) TBA  | 6) MODULATE NEW DATA
                MODULATION TYPE = 2   | ----------------------------------------
STARTING POINT IN DATA SEQUENCE = 100 |                        Option   = 2
THE NUMBER OF BITS DESIRED       = 400 |               PHASE REFERENCE = 45
THE NUMBER OF SAMPLES PER BIT    = 4   |FIRST BIT TO BE USED           = 1
ENTER IQ IMBALANCES? 1)Yes 2)No = 2   |NUM. OF BITS TO BE USED ( 2-398) = 50
                                      |                       Option   = 2
                BITS PER SYMBOL = 2   |
                                      |
                                      |
                                      |
                                      |
                                      |
                                      |
                                      |
--------------------------------------------------------------------------------
```

1/28/88 8:53 am

EYE PATTERNS



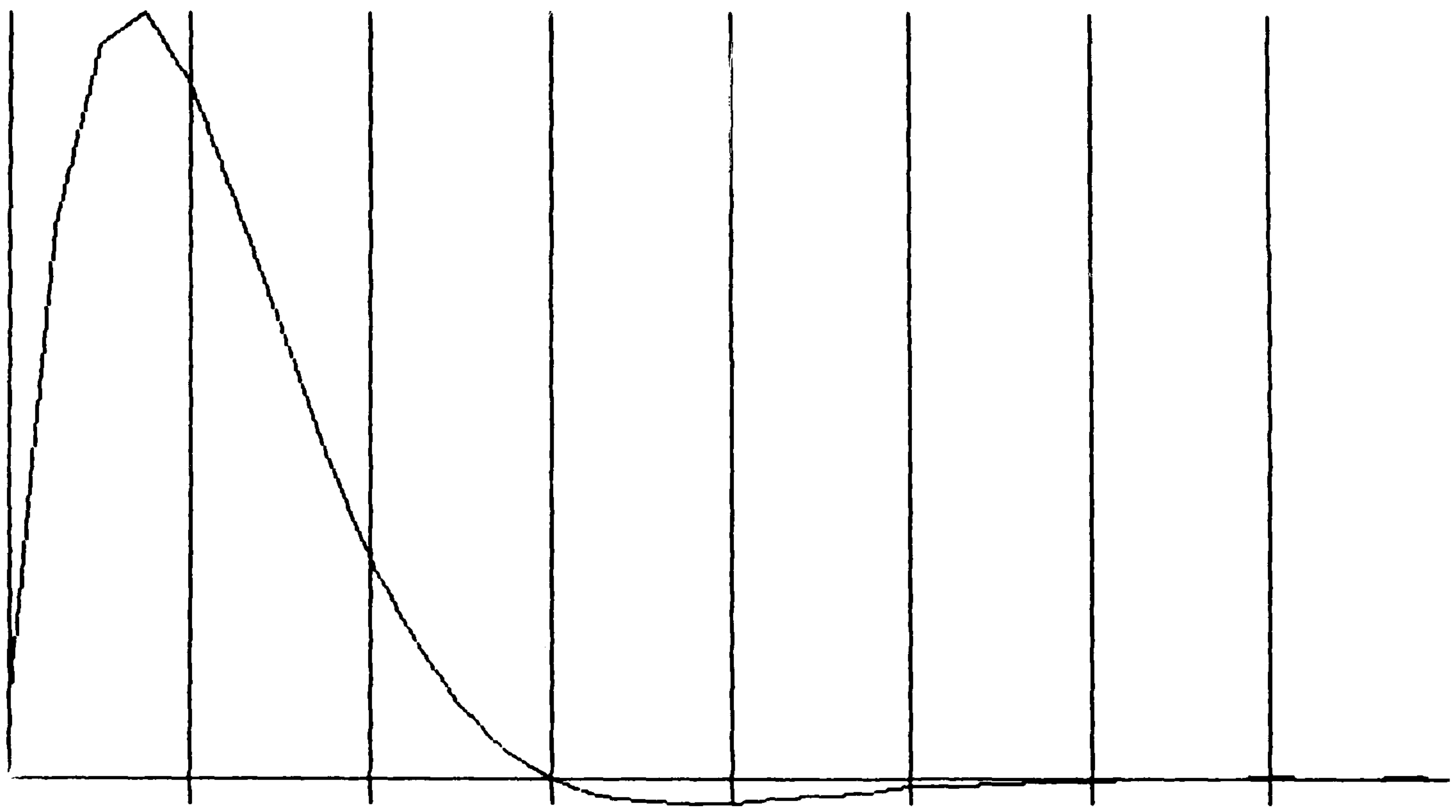
HORIZONTAL SCALE = 0.400 BITS/DIV.

F1
Forwd

```
1/28/88  8:53 am

--------------------------------------------------------------------------------
------------------------ MODEM  (c) 1987  by Carl Ryan -------------------------

   Current Signal =  in memory only
   MODEM contains the following functions:

          (1)    PN Generator
          (2)    Encoder
          (3)    Modulator
          (4)    Filter
          (5)    Transmitter
          (6)    Receiver
          (7)    Decoder/Demodulator/Monte Carlo
          (8)    Error Detection
          (9)    Time & Frequency Domain Plots
         (10)    Transversal/Recursive Adaptive Equalizer
         (11)    Signal Combiner
         (12)    Euclidian Distance Bounds
         (13)    Discriminator
         (14)    FSK Receiver
         (15)    Reconfigure Initialization for "MODEM"
         (16)    Utilities
   Enter the number of your choice (1-16) : 4

--------------------------------------------------------------------------------
```

```
1/28/88  8:53 am

---------------------------------------------------------------------------------
-- FILTER -- (c) 1986  BY CARL RYAN    ! Options :
                                       ! 1)  DISPLAY FREQUENCY RESPONSE
        *** Filter Types ***           ! 2)  DISPLAY IMPULSE RESPONSE
    1)    N POLE BUTTERWOR:H           ! 3)  CASCADE THE FILTER
    2)    N POLE CHEBYSHEV             ! 4)  BUILD NEW FILTER
    3)    IDEAL SHARP CUTOFF           !----------------------------------------
    4)    GAUSSIAN                     !                         Option = 2
    5)    MEASURED DATA                !
    6)    LOAD FILTER FROM DISK        !                         Option = 0
            FILTER TYPE DESIRED = 1    !
# BITS DURATION OF IMPULSE RESP.= 8    !
FULL SCALE FREQUENCY RESPONSE   = 5    !
# OF SAMPLES PER BIT PERIOD     = 4    !
       *** Filter Parameters ***       !
NORMALIZED RF BANDWIDTH         = 0.50!
NORMALIZED FREQUENCY OFFSET     = 0    !
 'MBER OF POLES                 = 2    !
 JALIZE GROUP DELAY 1)Yes 2)No  = 2    !
REFERENCE FREQUENCY             = 0    !
                                       !
                                       !
                                       !
---------------------------------------------------------------------------------
```

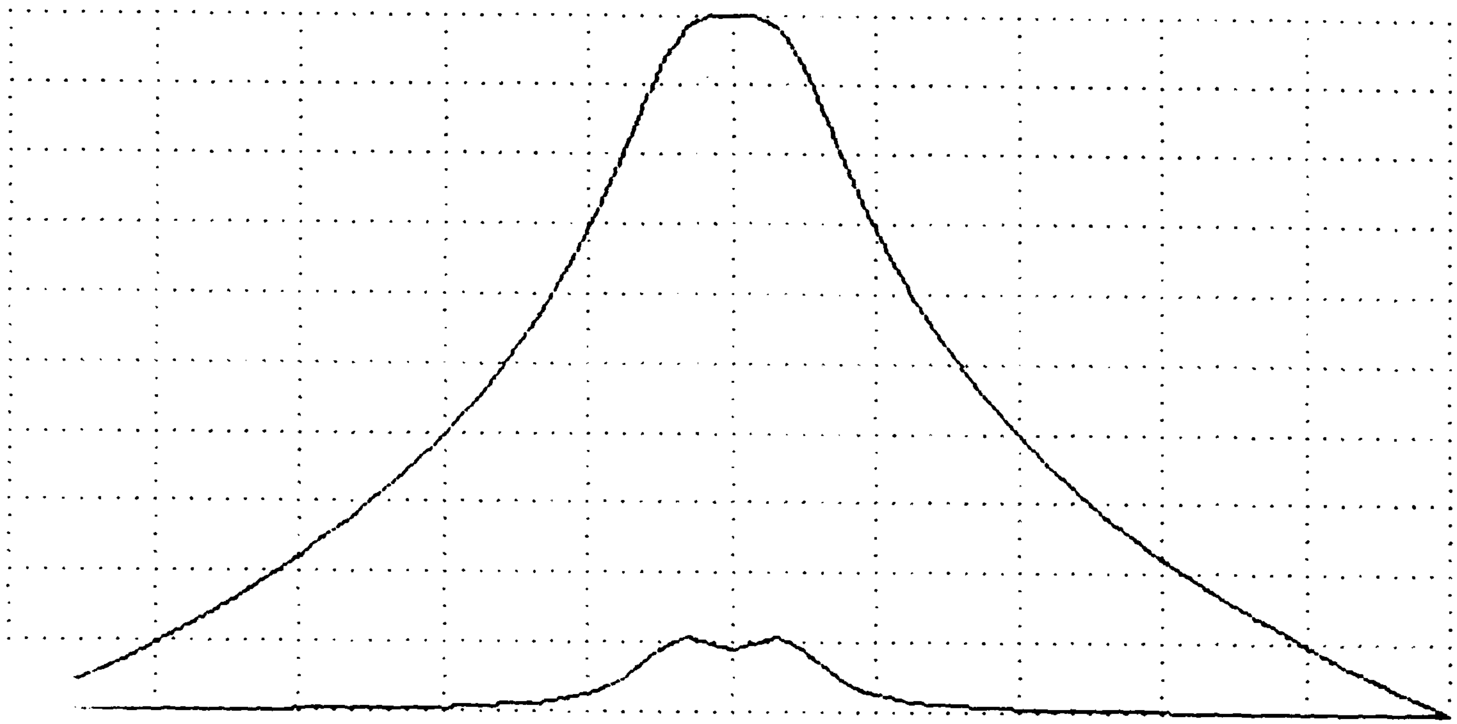
1/28/88 8:53 am



IMPULSE RESPONSE: NOISE BANDWIDTH = 0.555

Forwd MAXIMUM FREQUENCY GAIN = 0.000 dB AT FREQUENCY 0.000 R

1/28/88 8:54 am



FREQUENCY RESPONSE: MAX VALUE = 0.000 dB AT FREQUENCY 0.000 R
HORIZONTAL = 0.50 R PER DIV. NOISE BANDWIDTH = 0.555
F1 VERTICAL: GROUP DELAY= 1/R PER DIV. FREQUENCY= 4.000 dB PER DIV.
Forwd

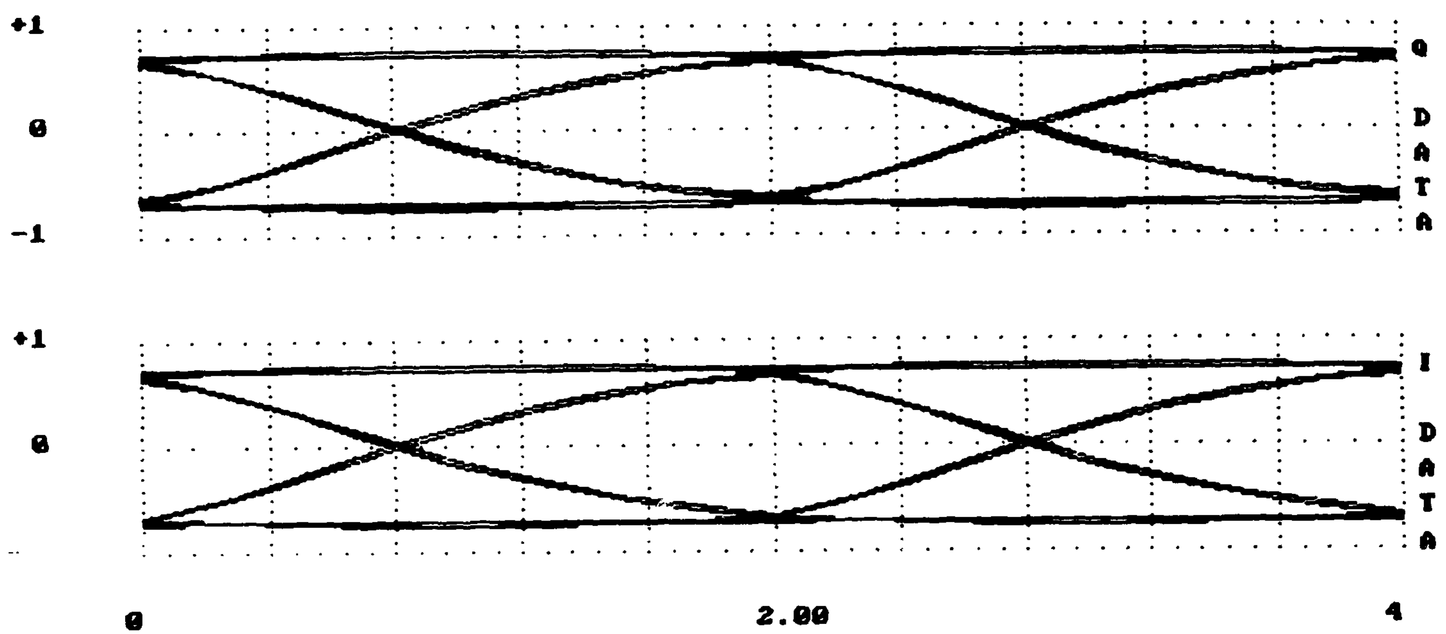
```
1/28/88  8:54 am

----------------------------------------------------------------------------
------------------------ MODEM  (c) 1987  by Carl Ryan ----------------------

   Current Signal =  in memory only   Current Filter =  in memory only
   MODEM contains the following functions:

         (1)    PN Generator
         (2)    Encoder
         (3)    Modulator
         (4)    Filter
         (5)    Transmitter
         (6)    Receiver
         (7)    Decoder/Demodulator/Monte Carlo
         (8)    Error Detection
         (9)    Time & Frequency Domain Plots
        (10)    Transversal/Recursive Adaptive Equalizer
        (11)    Signal Combiner
        (12)    Euclidian Distance Bounds
        (13)    Discriminator
        (14).   FSK Receiver
        (15)    Reconfigure Initialization for "MODEM"
        (16)    Utilities
   Enter the number of your choice (1-16) : 5

----------------------------------------------------------------------------
```

1/28/88 8:55 am

```
---------------------------------------------------------------------------
-- TRANSMITTER -- (c)1986 BY CARL RYAN!OPTIONS:
                                     ! 1)  PLOT ENVELOPE AMPLITUDE
------------      ------------       ! 2)  PLOT EYE PATTERNS
!  SIGNAL  ! -----> !  FILTER  !      ! 3)  PLOT "I" AND "Q"  DATA
------------      ------------       ! 4)  PLOT SIGNAL CONSTELLATION
                                     ! 5)  PLOT EXCESS PHASE
Use  Signal in memory?  1)Yes 2)No = 1! 6)  SELECT NEW SIGNAL OR NEW FILTER
Signal =  in memory only              ! 7)  TRANSMITTER NONLINEARITY
  Filtering the signal is optional.   ! (NOTE: Data saved will be Linear)
  Filter the signal?    1)Yes 2)No = 1!--------------------------------------
Use  Filter in memory?  1)Yes 2)No = 1!                      Option    = 2
Filter =  in memory only              !             PHASE REFERENCE = 45
                                      !FIRST BIT TO BE USED          = 1
                                      !NUM. OF BITS TO BE USED ( 2-398) = 100
                                      !                      Option    = 0
                                      !
                                      !
                                      !
                                      !
                                      !
                                      !
                                      !
---------------------------------------------------------------------------
```

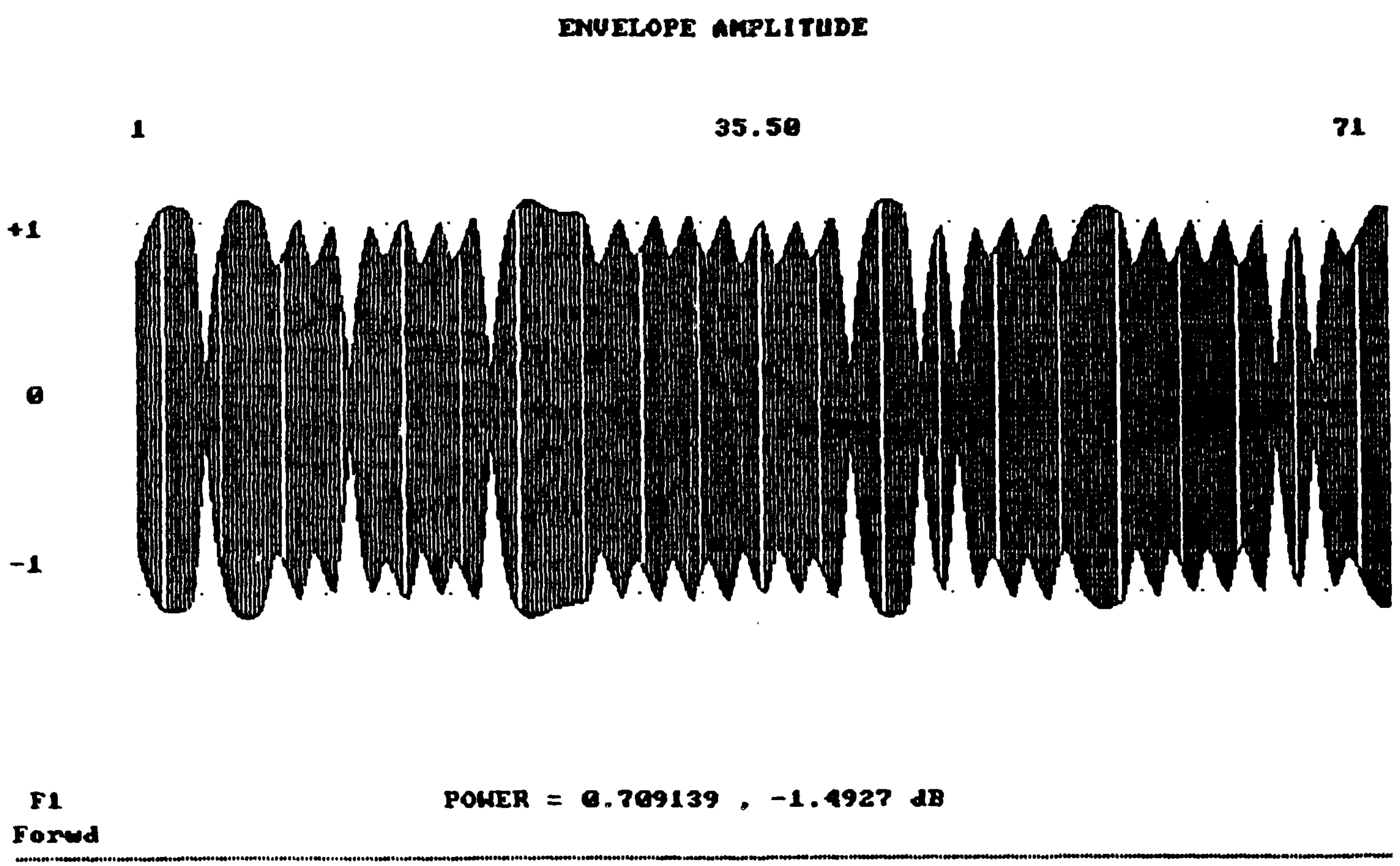
1/28/88 8:55 am

EYE PATTERNS



F1
For~d

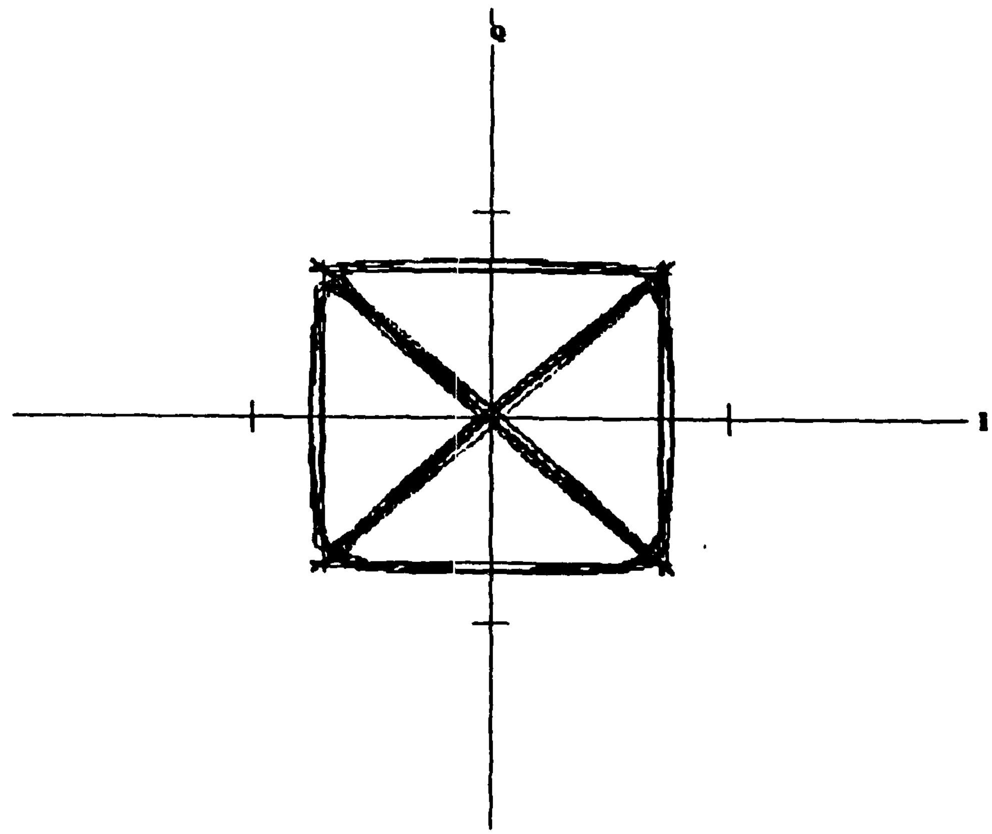
1/28/88 8:56 am



ENVELOPE AMPLITUDE
1
35.50
71
+1
0
-1
F1
Forwd
POWER = 0.709139 , -1.4927 dB

1/28/88 8:56 am

SIGNAL CONSTELLATION



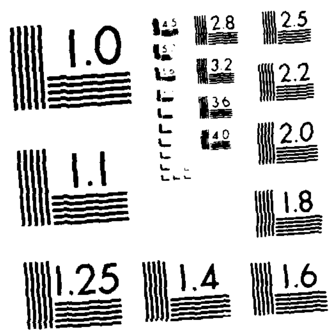
F1
Forwd

1/28/88 8:57 am

```
-------------------------------------------------------------------------------
-- TRANSMITTER -- (c)1986 BY CARL RYAN!OPTIONS:
                                      ! 1)  PLOT ENVELOPE AMPLITUDE
------------      ------------        ! 2)  PLOT EYE PATTERNS
! SIGNAL  ! -----> !  FILTER  !        ! 3)  PLOT "I" AND "Q"  DATA
------------      ------------        ! 4)  PLOT SIGNAL CONSTELLATION
                                      ! 5)  PLOT EXCESS PHASE
Use  Signal in memory?  1)Yes 2)No = 1! 6)  SELECT NEW SIGNAL OR NEW FILTER
Signal =  in memory only              ! 7)  TRANSMITTER NONLINEARITY
  Filtering the signal is optional.   ! (NOTE: Data saved will be Linear)
  Filter the signal?    1)Yes 2)No = 1!----------------------------------------
Use  Filter in memory?  1)Yes 2)No = 1!-> PREDEFINED NONLINEARITY
Filter =  in memory only              !EQUATION IS :
                                      !Y =  B*X + C*X^2 + D*X^3 + E*X^4
                                      !Coefficients used :
                                      !Gain :  B = 1.7 C = -0.4 D = -0.3 E = 0.0
                                      !
                                      !Phase shift :
                                      !A = 3.0 B = 0.0 C = 20.0
                                      !D = 59.0 E = -30.0
                                      !        BACK OFF LEVEL (IN DB) = 0
                                      !
                                      !
-------------------------------------------------------------------------------
```

AD-A194 336 ADVANCED COMMUNICATION SYSTEM ENGINEERING PROCEEDINGS 5/5
WORKSHOP HELD IN SE..(U) UNIVERSITY OF SOUTHERN
CALIFORNIA LOS ANGELES COMMUNICATION S.. R A SCHOLTZ
UNCLASSIFIED MAR 88 CSI-88-03-02 ARO-24913.1-EL-CF F/G 25/5 NL

END
DATE
FILMED
8 88



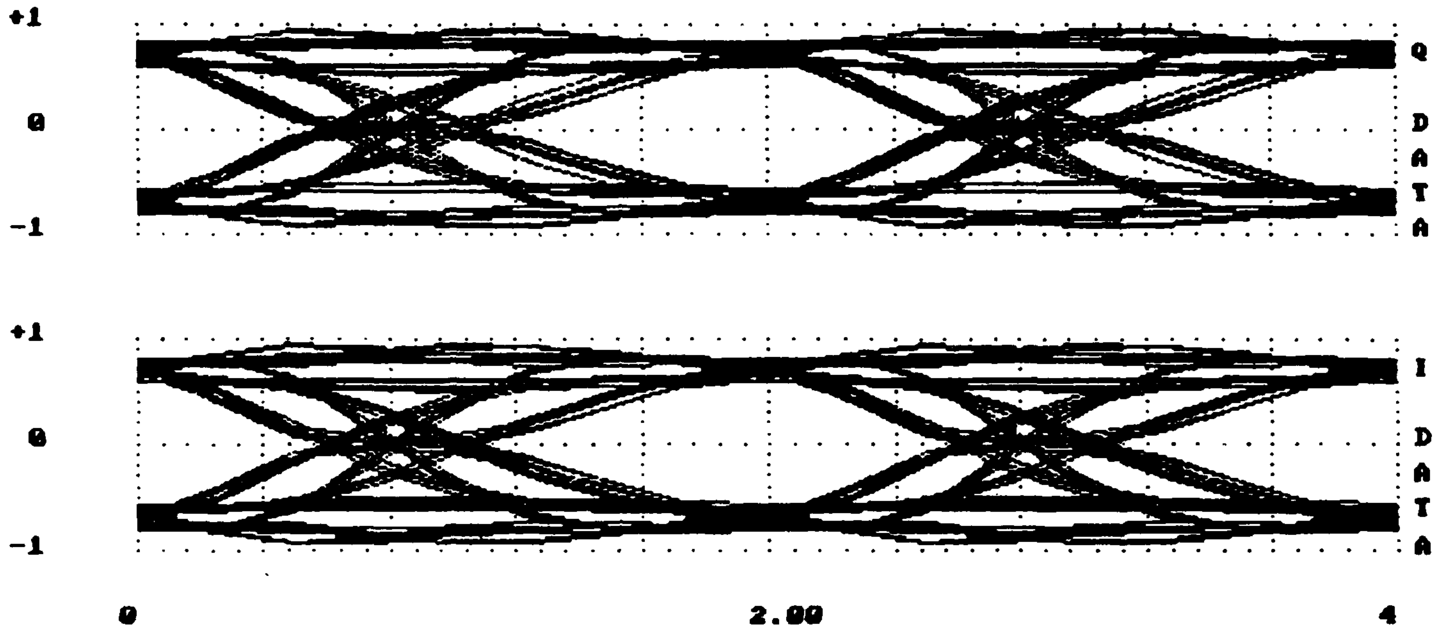
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

1/28/88 8:59 am

```
-----------------------------------------------------------------------------------
-- TRANSMITTER -- (c)1986 BY CARL RYAN!OPTIONS:
                                      ! 1)  PLOT ENVELOPE AMPLITUDE
------------       ------------       ! 2)  PLOT EYE PATTERNS
!  SIGNAL  ! -----> !  FILTER  !      ! 3)  PLOT "I" AND "Q"  DATA
------------       ------------       ! 4)  PLOT SIGNAL CONSTELLATION
                                      ! 5)  PLOT EXCESS PHASE
Use  Signal in memory?  1)Yes 2)No = 1! 6)  SELECT NEW SIGNAL OR NEW FILTER
Signal =  in memory only              ! 7)  TRANSMITTER NONLINEARITY
  Filtering the signal is optional.   ! (NOTE: Data saved will be NonLinear)
  Filter the signal?    1)Yes 2)No = 1!----------------------------------------
Use  Filter in memory?  1)Yes 2)No = 1!                          Option  = 2
Filter =  in memory only              !                 PHASE REFERENCE = 0
                                      !FIRST BIT TO BE USED              = 10
                                      !NUM. OF BITS TO BE USED ( 2-388) = 193
                                      !                          Option  = 0
                                      !
                                      !
                                      !
                                      !
                                      !
                                      !
                                      !
-----------------------------------------------------------------------------------
```

1/28/88 8:59 am

EYE PATTERNS

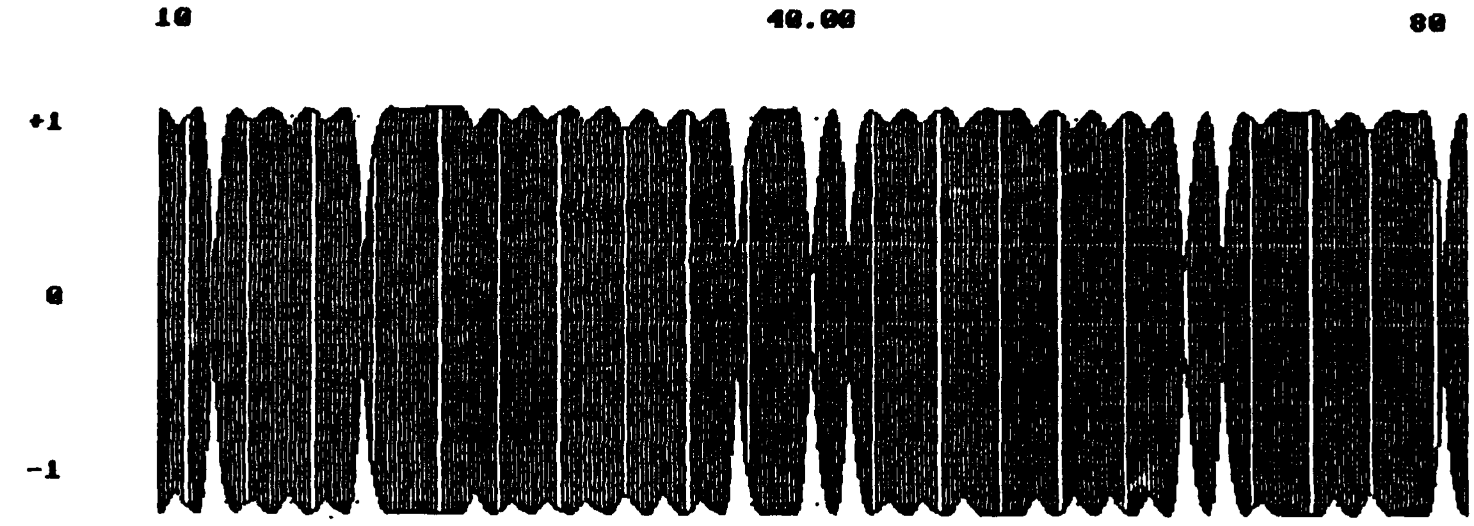


HORIZONTAL SCALE = 0.400 BITS/DIV.

F1
Forwd

1/28/88 8:59 am

ENVELOPE AMPLITUDE

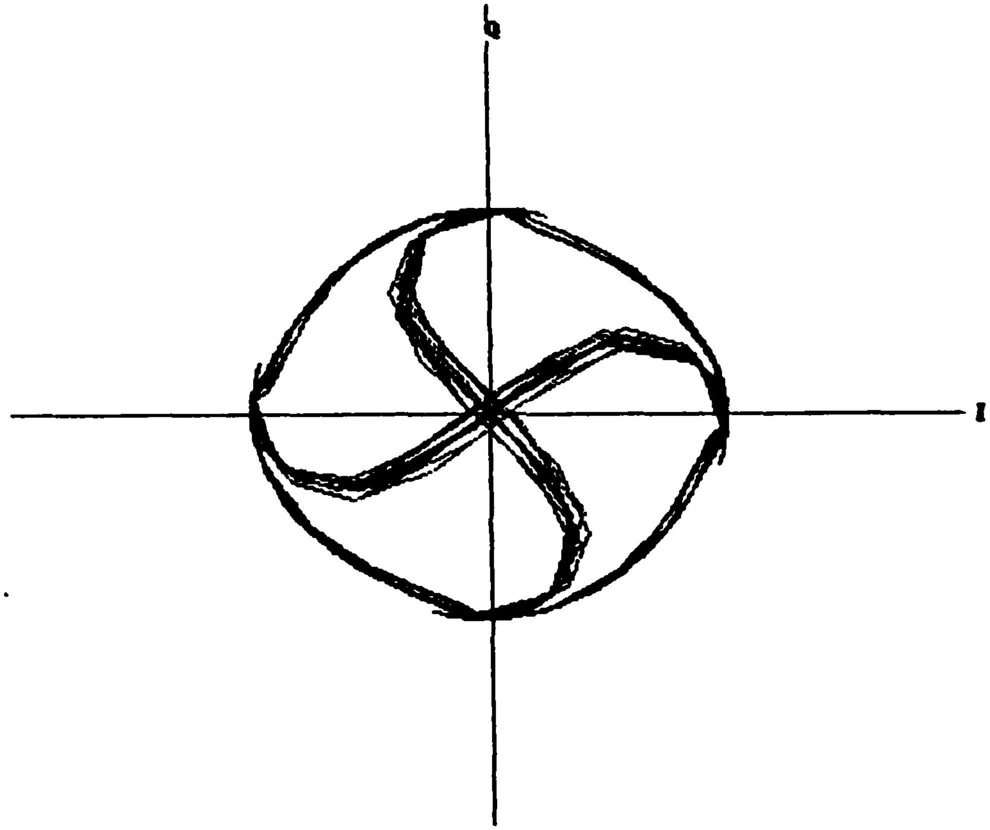


F1
Forwd

POWER = 0.859460 , -0.6577 dB

1/28/88 9:00 am

SIGNAL CONSTELLATION



F1
Forwd

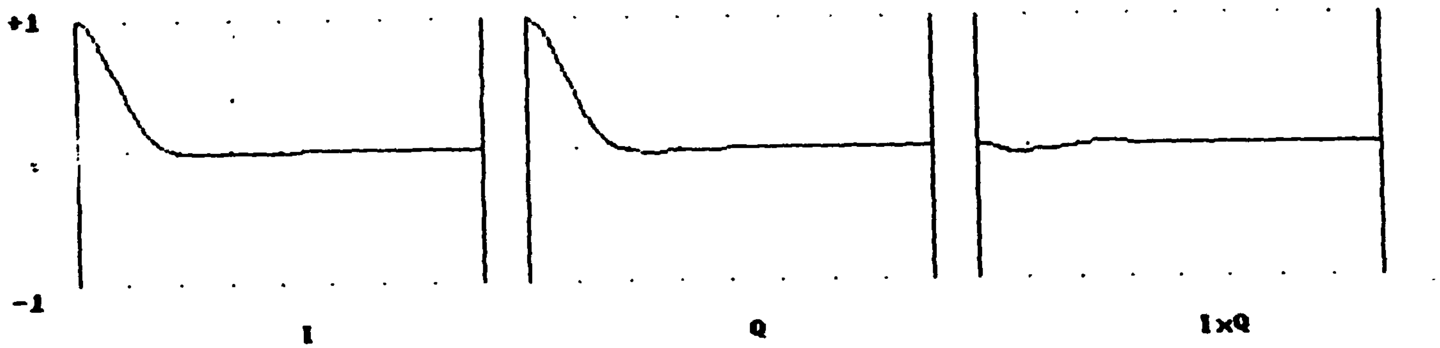
1/28/88 9:00 am

```
-------------------------------------------------------------------------------
----------------------- MODEM  (c) 1987  by Carl Ryan ---------------------

  Current Signal =  in memory only   Current Filter =  in memory only
  MODEM contains the following functions:

        (1)   PN Generator
        (2)   Encoder
        (3)   Modulator
        (4)   Filter
        (5)   Transmitter
        (6)   Receiver
        (7)   Decoder/Demodulator/Monte Carlo
        (8)   Error Detection
        (9)   Time & Frequency Domain Plots
       (10)   Transversal/Recursive Adaptive Equalizer
       (11)   Signal Combiner
       (12)   Euclidian Distance Bounds
       (13)   Discriminator
       (14)   FSK Receiver
       (15)   Reconfigure Initialization for "MODEM"
       (16)   Utilities
  Enter the number of your choice (1-16) : 9

-------------------------------------------------------------------------------
```

```
1/28/88  9:01 am

------------------------------------------------------------------------------
   -- Time & Frequency Domain Plots -- (c) 1986  by  Carl Ryan
      Current signal =  in memory only

       TIME PLOTS                               FREQUENCY PLOTS
 1) ENVELOPE AMPLITUDE & POWER               7) POWER SPECTRAL DENSITY (PSD)
 2) EYE DIAGRAM                              8) I PSD
 3) I & Q DATA                               9) Q PSD
 4) SIGNAL CONSTELLATION                    10) IxQ PSD
 5) EXCESS PHASE                            11) OUT OF BAND POWER
 6) HISTOGRAM                               12) AUTO CORRELATIONS
------------------------------------------------------------------------------
                              Option = 12  !USE ENTIRE SIGNAL?  1)YES 2)NO  = 2
   DURATION OF AUTO-CORRELATION  = 12  !                     STARTING BIT = 4
        ( in bit times )               !                   NUMBER OF BITS = 256
    PHASE REFERENCE (in degrees) = 0   !                           Option = 12
  1)RECTANGULAR 2)RAISED COSINE        !
                     WINDOW TYPE = 2   !
                                       !
                                       !
                                       !
                                       !

------------------------------------------------------------------------------
```

1/28/88 9:01 am

Auto-Correlations



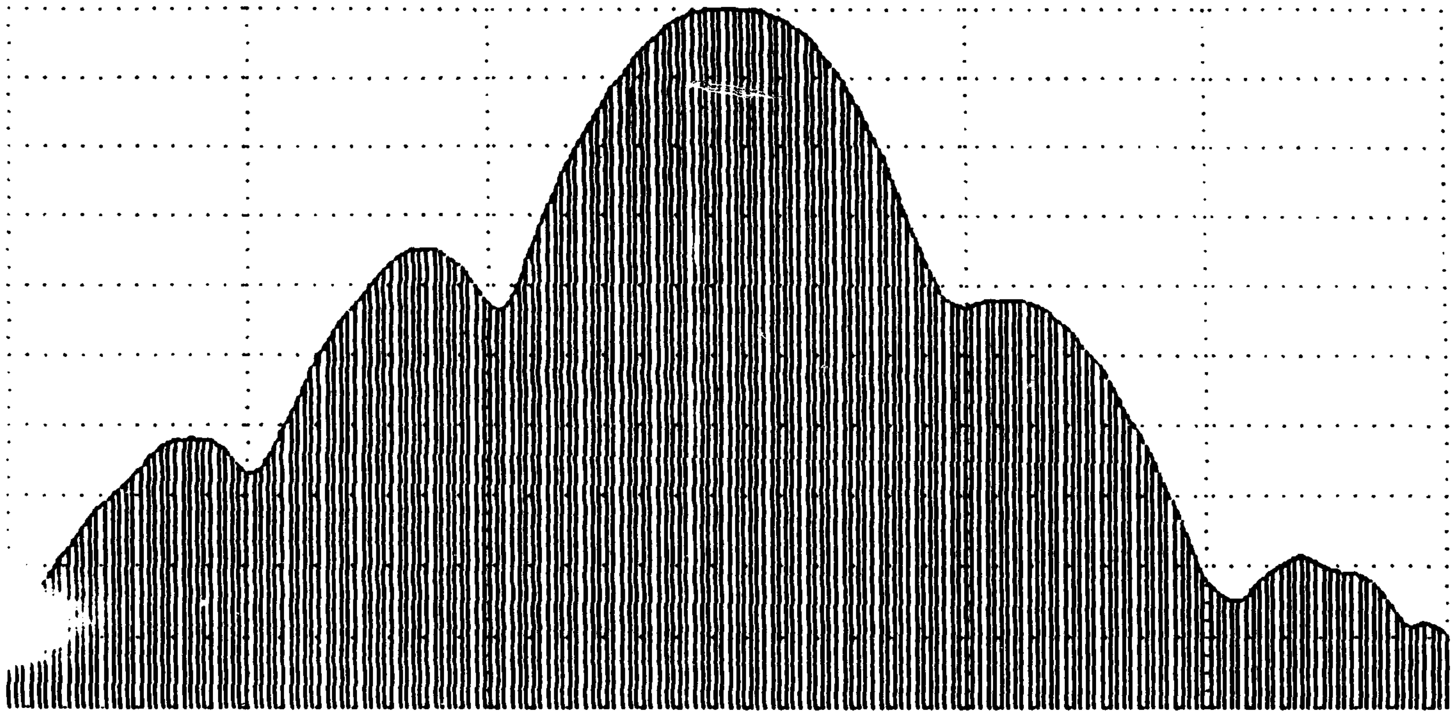
HORIZONTAL = 1.5000 BITS PER DIVISION

F1
Forwd

1/28/88 9:01 am

```
--------------------------------------------------------------------------------
   -- Time & Frequency Domain Plots -- (c) 1986  by  Carl Ryan
      Current signal =  in memory only

        TIME PLOTS                              FREQUENCY PLOTS
 1) ENVELOPE AMPLITUDE & POWER               7) POWER SPECTRAL DENSITY (PSD)
 2) EYE DIAGRAM                              8) I PSD
 3) I & Q DATA                               9) Q PSD
 4) SIGNAL CONSTELLATION                    10) IxQ PSD
 5) EXCESS PHASE                            11) OUT OF BAND POWER
  ) HISTOGRAM                               12) AUTO CORRELATIONS
--------------------------------------------------------------------------------
                           Option = 7   !      SHADE GRAPH ?  1)YES 2)NO = 1
     FULL SCALE FREQUENCY RANGE = 3     ! SPACING BETWEEN VERTICAL GRIDS = 0.500
          FREQUENCY RESOLUTION = 0.03!      FULL SCALE VERTICAL IN dB = 50
 1)RECTANGULAR 2)RAISED COSINE          !       PLOT FCC MASK 1)YES 2)NO = 2
                     WINDOW TYPE = 2    !    STORE LAST PLOT? 1)YES 2)NO = 2
                                        !                         Option = 7
                                        !
                                        !
                                        !
                                        !

--------------------------------------------------------------------------------
```

1/28/88 9:02 am



POWER SPECTRAL DENSITY

F1 HORIZONTAL = 0.5000 R PER DIV. VERTICAL = 5.0000 dB PER DIV.

Forwd PSD(max) = 0.2476 dB

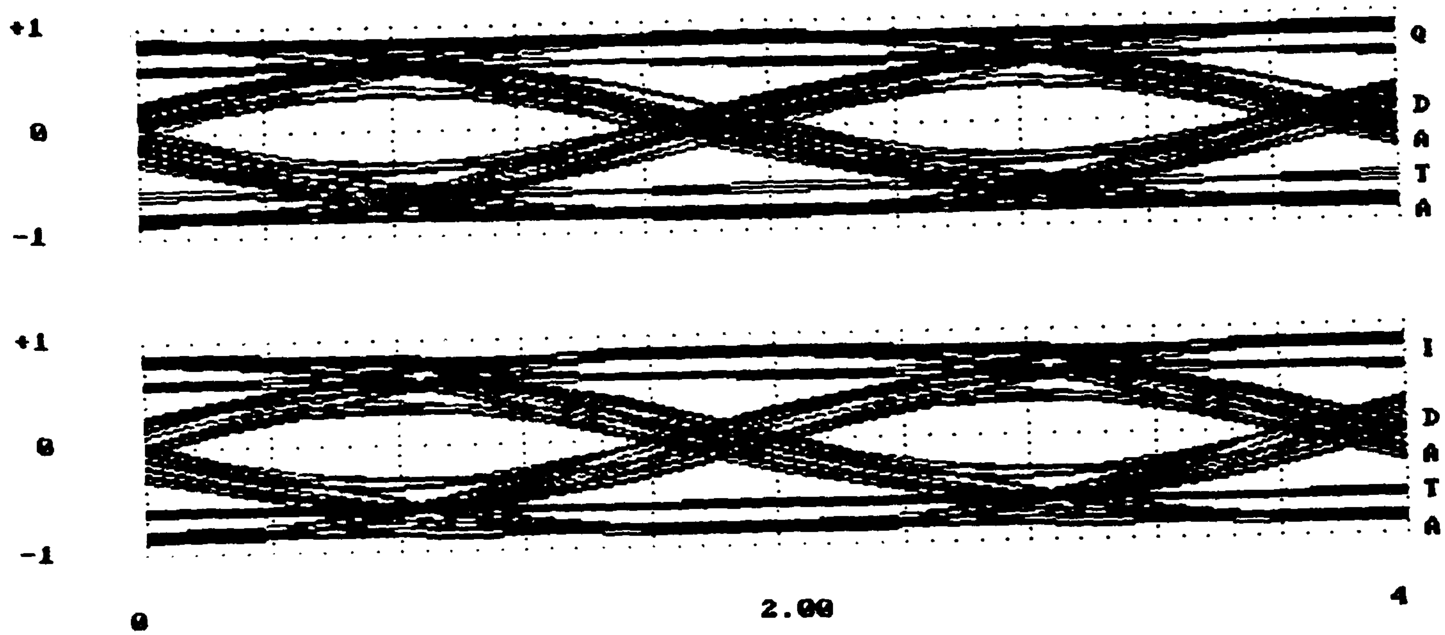
```
1/28/88  9:02 am

------------------------------------------------------------------------------
------------------------ MODEM  (c) 1987  by Carl Ryan -----------------------

  Current Signal =  in memory only   Current Filter =  in memory only
  MODEM contains the following functions:

        (1)    PN Generator
        (2)    Encoder
        (3)    Modulator
        (4)    Filter
        (5)    Transmitter
        (6)    Receiver
        (7)    Decoder/Demodulator/Monte Carlo
        (8)    Error Detection
        (9)    Time & Frequency Domain Plots
       (10)    Transversal/Recursive Adaptive Equalizer
       (11)    Signal Combiner
       (12)    Euclidian Distance Bounds
       (13)    Discriminator
       (14)    FSK Receiver
       (15)   .Reconfigure Initialization for "MODEM"
       (16)    Utilities
  Enter the number of your choice (1-16) : 6

------------------------------------------------------------------------------
```

1/28/88 9:04 am

```
--------------------------------------------------------------------------------
-- RECEIVER -- (c)1986 BY CARL RYAN   !OPTIONS:
                                      ! 1)  PLOT ENVELOPE AMPLITUDE
------------        ------------      ! 2)  PLOT EYE PATTERNS
!  SIGNAL  ! -----> !  FILTER  !      ! 3)  PLOT "I" AND "Q"  DATA
------------        ------------      ! 4)  PLOT SIGNAL CONSTELLATION
                                      ! 5)  PLOT EXCESS PHASE
Use  Signal in memory?  1)Yes 2)No = 1! 6)  SELECT NEW SIGNAL OR NEW FILTER
Signal =  in memory only              ! (NOTE: Data saved will be Linear)
  Filtering the signal is optional.   !----------------------------------------
  Filter the signal?    1)Yes 2)No = 1!
Use  Filter in memory?  1)Yes 2)No = 1!                          Option  = 2
Filter =  in memory only              !                 PHASE REFERENCE = 0
                                      !FIRST BIT TO BE USED             = 1
                                      !NUM. OF BITS TO BE USED ( 2-398) = 198
                                      !                          Option  = 0
                                      !
                                      !
                                      !
                                      !
                                      !
                                      !
                                      !
--------------------------------------------------------------------------------
```

1/28/88 9:04 am

EYE PATTERNS



HORIZONTAL SCALE = 0.400 BITS/DIV.

F1
Forwd

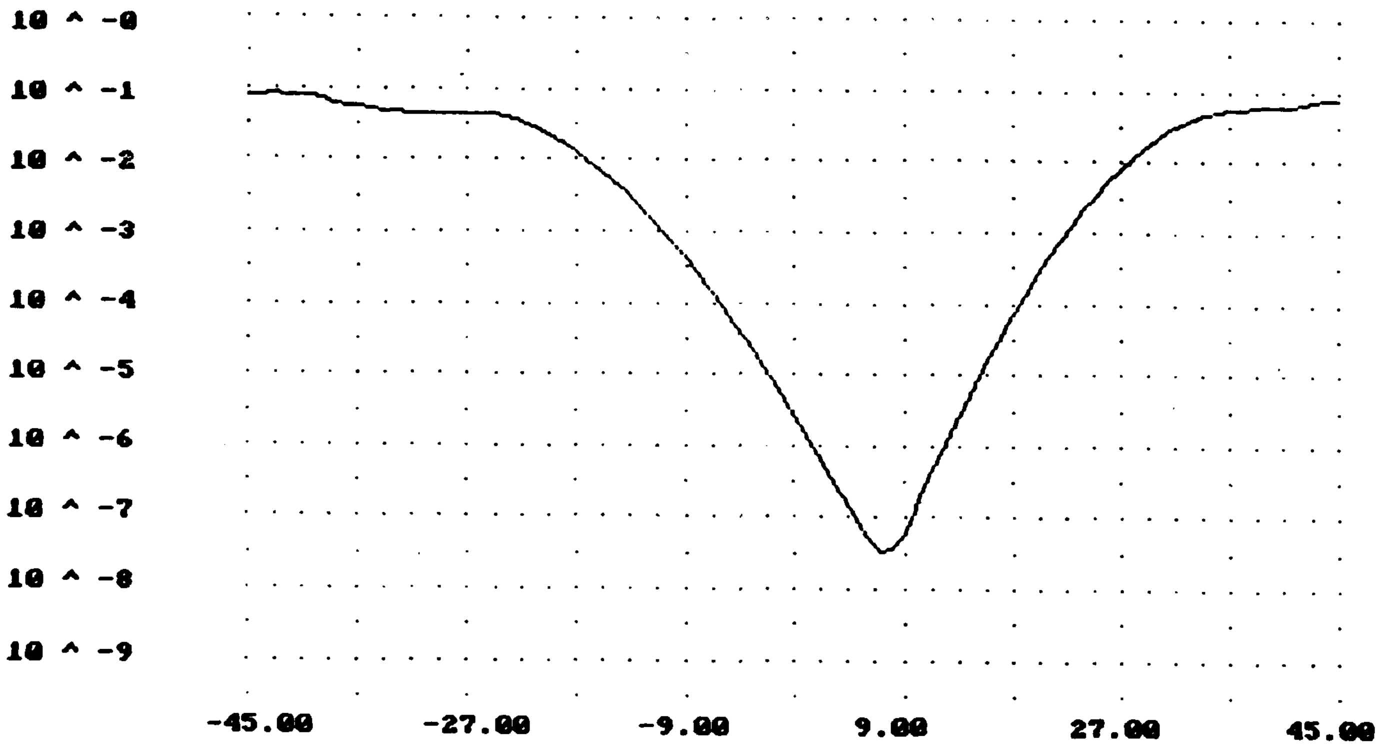
```
1/28/88  9:04 am

---------------------------------------------------------------------------
------------------------ MODEM  (c) 1987  by Carl Ryan ---------------------

  Current Signal =  in memory only   Current Filter =  in memory only
  MODEM contains the following functions:

        (1)   PN Generator
        (2)   Encoder
        (3)   Modulator
        (4)   Filter
        (5)   Transmitter
        (6)   Receiver
        (7)   Decoder/Demodulator/Monte Carlo
        (8)   Error Detection
        (9)   Time & Frequency Domain Plots
       (10)   Transversal/Recursive Adaptive Equalizer
       (11)   Signal Combiner
       (12)   Euclidian Distance Bounds
       (13)   Discriminator
       (14)   FSK Receiver
       (15)   Reconfigure Initialization for "MODEM"
       (16)   Utilities
  Enter the number of your choice (1-16) : 8

---------------------------------------------------------------------------
```

```
1/28/88   9:33 am

---------------------------------------------------------------------------
-- ERROR ANALYSIS -- (c) by CARL RYAN


Use  Signal in memory?  1)Yes 2)No = 1
        NOISE BANDWIDTH OF SIGNAL = 0.5
55
   ENTER NEW BANDWIDTH? 1)Yes 2)No = 2

---------------------------------------------------------------------------
```

1/28/88 9:36 am

```
-----------------------------------------------------------------------------
-- ERROR ANALYSIS -- (c) by CARL RYAN |BIT ERROR RATE OPTIONS
   Signal =   in memory only          |  1) I BER GRAPH
--------------------------------------|  2) Q BER GRAPH
      NUMBER OF BITS TO BE USED = 200 |  3) AVERAGE BER GRAPH
                                      |  4) LIST BER VALUES to SCREEN
        VARIABLE PARAMETERS           |  5) LIST BER VALUES to PRINTER
1)  Eb/No        3) Phase Reference   |  6) EXIT BER MENU
2)  Sample Time  4) Decision Threshold|-------------------------------------
 Parameter to vary             = 3    |
 1) Error Function 2) Exponetial      |Option                      = 3
 3) DPSK = 1                          |   Store Last Plot ? 1)YES 2)NO = 2
     Eb/No STARTING POINT IN dB = 16  |
 PHASE STARTING POINT (IN DEG.) = -45 |
                          RANGE = 90  |
          ENTER THE RESOLUTION = 1    |
   I SAMPLE TIME STARTING POINT = .8  |
   Q SAMPLE TIME STARTING POINT = 0.80|
     Duration Between Samples   = 2   |
 DECISION THRESHOLD START POINT = 0   |
Use ERROR CORRECTION DECODING ?       |
                    1)YES 2)NO = 2    |
                                      |
-----------------------------------------------------------------------------
```

1/28/88 9:37 am



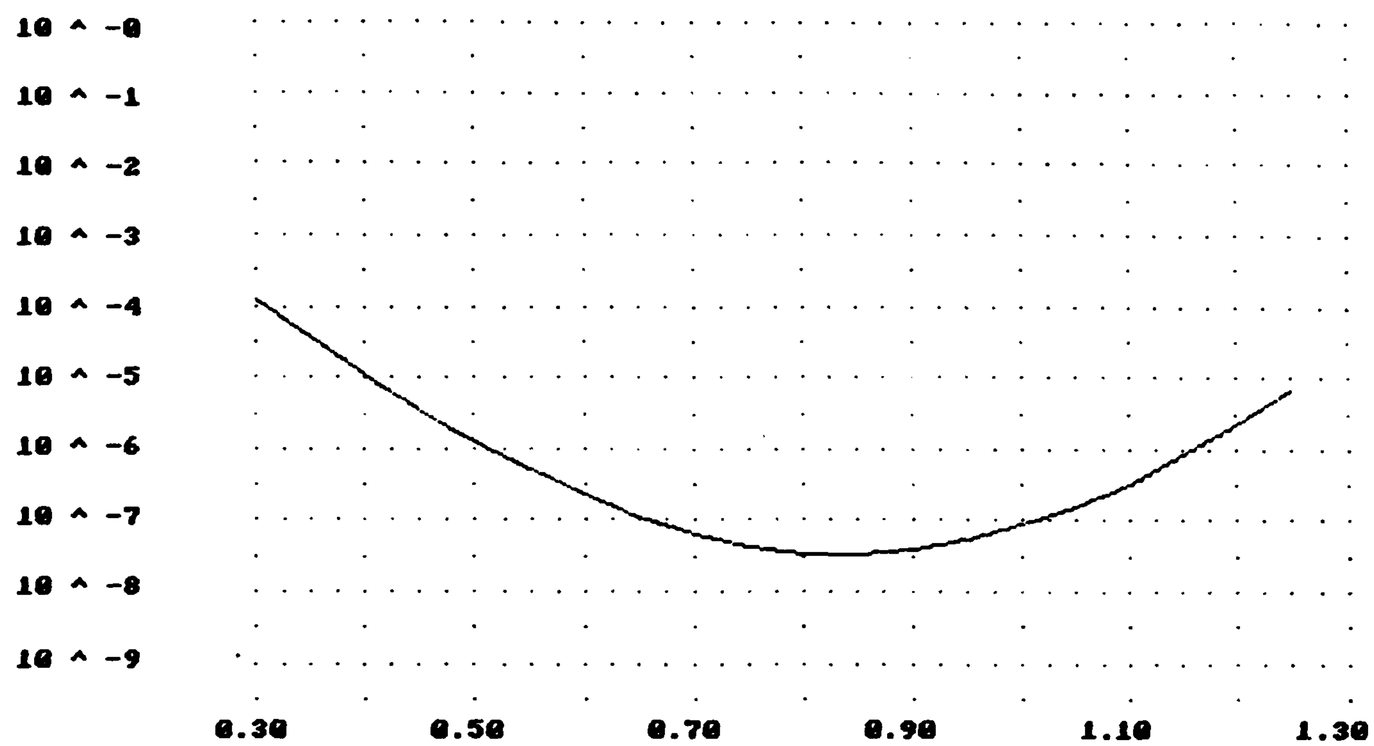
Eb/No = 16.00
SAMPLE TIME = 0.80
F1 DECISION THRESHOLD = 0.00
Forwd

BIT ERROR RATE
vs.
PHASE REFERENCE (degrees)

1/28/88 9:40 am

```
---------------------------------------------------------------------------------
-- ERROR ANALYSIS -- (c) by CARL RYAN |BIT ERROR RATE OPTIONS
   Signal =   in memory only          |  1) I BER GRAPH
--------------------------------------|  2) Q BER GRAPH
      NUMBER OF BITS TO BE USED = 200 |  3) AVERAGE BER GRAPH
                                      |  4) LIST BER VALUES to SCREEN
        VARIABLE PARAMETERS           |  5) LIST BER VALUES to PRINTER
1)  Eb/No        3) Phase Reference   |  6) EXIT BER MENU
2)  Sample Time  4) Decision Threshold|------------------------------------
 Parameter to vary              = 2   |
 1). Error Function 2) Exponetial     |Option                            = 3
 3) DPSK = 1                          |   Store Last Plot ? 1)YES 2)NO = 2
     Eb/No STARTING POINT IN dB = 16  |
 PHASE STARTING POINT (IN DEG.) = 7   |
   I SAMPLE TIME STARTING POINT = .3  |
   Q SAMPLE TIME STARTING POINT = 0.30|
     Duration Between Samples   = 2   |
                          RANGE = 1   |
            ENTER THE RESOLUTION = .05|
 DECISION THRESHOLD START POINT = 0   |
Use ERROR CORRECTION DECODING ?       |
                  1)YES 2)NO = 2      |
                                      |

---------------------------------------------------------------------------------
```

1/28/88 9:40 am



Eb/No = 16.00
PHASE REFERENCE = 7.00
DECISION THRESHOLD = 0.00

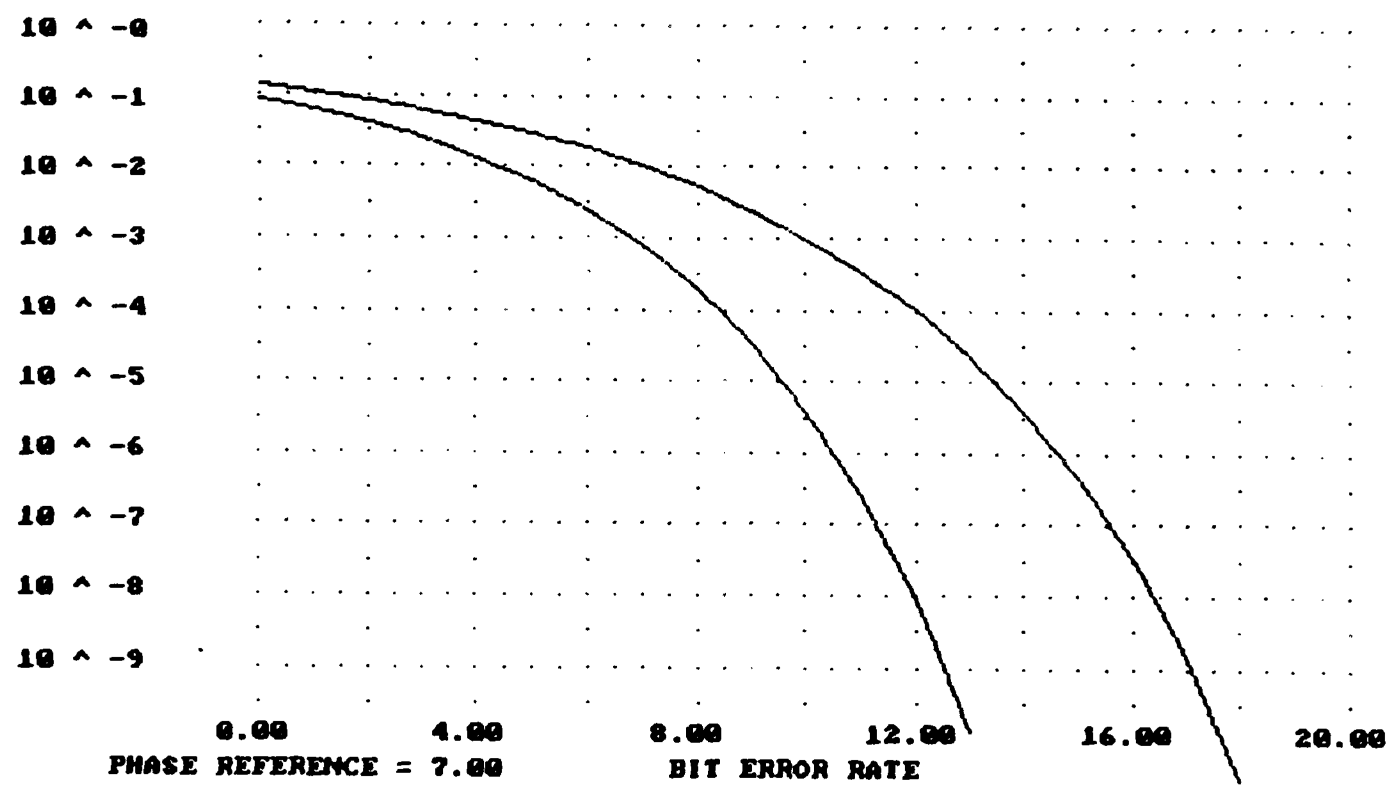
BIT ERROR RATE
vs.
SAMPLE TIME (bit times)

Forwd

1/28/88 9:41 am

```
---------------------------------------------------------------------------------
-- ERROR ANALYSIS -- (c) by CARL RYAN !BIT ERROR RATE OPTIONS
   Signal =   in memory only          !  1) I BER GRAPH
--------------------------------------!  2) Q BER GRAPH
      NUMBER OF BITS TO BE USED = 200 !  3) AVERAGE BER GRAPH
                                      !  4) LIST BER VALUES to SCREEN
        VARIABLE PARAMETERS           !  5) LIST BER VALUES to PRINTER
1)  Eb/No         3) Phase Reference  !  6) EXIT BER MENU
2)  Sample Time  4) Decision Threshold!---------------------------------------
 Parameter to vary                = 1 !
 1) Error Function 2) Exponetial      !Option                            = 3
 3) DPSK = 1                          !(1) BPSK
     Eb/No STARTING POINT IN dB = 0   !(2) 8PSK
                          RANGE = 20  !(3) 16PSK
           ENTER THE RESOLUTION = 1   !(4) CPFSK  m=4, h=1/4 2REC
 PHASE STARTING POINT (IN DEG.) = 7   !(5) CPFSK  m=8, h=1/8 3REC
   I SAMPLE TIME STARTING POINT = .8  !(6) no ideal curve
   Q SAMPLE TIME STARTING POINT = 0.80!Select Ideal Curve ..: 1
     Duration Between Samples   = 2   !   Store Last Plot ? 1)YES 2)NO = 2
 DECISION THRESHOLD START POINT = 0   !
Use ERROR CORRECTION DECODING ?       !
                   1)YES 2)NO = 2     !
                                      !

---------------------------------------------------------------------------------
```

1/28/88 9:41 am



PHASE REFERENCE = 7.00
SAMPLE TIME = 0.80
F1 DECISION THRESHOLD = 0.00
Forwd

BIT ERROR RATE
vs.
Eb/No

```
----------------------- MODEM  (c) 1987  by Carl Ryan ----------------------

  Current Signal =  in memory only   Current Filter =  in memory only
  MODEM contains the following functions:

        (1)    PN Generator
        (2)    Encoder
        (3)    Modulator
        (4)    Filter
        (5)    Transmitter
        (6)    Receiver
        (7)    Decoder/Demodulator/Monte Carlo
        (8)    Error Detection
        (9)    Time & Frequency Domain Plots
       (10)    Transversal/Recursive Adaptive Equalizer
       (11)    Signal Combiner
       (12)    Euclidian Distance Bounds
       (13)    Discriminator
       (14)    FSK Receiver
       (15)    Reconfigure Initialization for "MODEM"
       (16)    Utilities
  Enter the number of your choice (1-16) :

                          Hit [C] to CANCEL EXIT
                          Strike any key to exit
```

path coefficients, if you choose you can try to make it converge too rapidly, you'll get bad results. This is the algorithm I'm using: So long as it's relative slow convergence, in other words, long time constants, it works fine.

MOHANTY: Okay, thank you.

CHOMA: Thank you Carl. There is a photographer waiting outside and since we're running a little bit behind schedule I guess we should go outside and get our group picture taken. Bob ....

SCHOLTZ: Yes, we'd like to have sort of a memorial photograph so if you see anyone else around here who belongs in our group, just pull them in. The photographer would like to take the picture over in the patio in front of the restaurant, in that area. So if you'll just wander over that way we'll get a group picture. Thank you.

USC-CSI WORKSHOP ON ADVANCED COMMUNICATION SYSTEM ENGINEERING

Proceedings of Session Five:
A Group Discussion: The Role of Government, Industry, and Universities in Advanced Communication System Engineering

GOLOMB: In our panel today we have Bill Sander from the Army Research Office who is representing government, I suppose, and then Dick Booton of TRW representing industry, and on my left Ray Pickholtz of George Washington University representing the academic community. I thought perhaps the best way to begin would be to ask each of them to make a few introductory remarks of, I don't know, up to 5 or 10 minutes if they want to talk that long. Why don't I start on my left here, if you're willing to start this off ....

PICKHOLTZ: Hello, I'll keep my remarks brief. I've sat through two days of a workshop, the title of which was "Advanced Communication Systems Engineering," but most of what I heard was on computer aided design, and it's not clear to me that the two are equivalent. I think computer aided design is very important but I think there's a danger of becoming complacent. In fact it reminds me of a story I heard from Phil Balaban at lunch on the first day -- it's actually a joke about this hospital orderly that was giving injections to a group of patients in a clinic using the same needle. When somebody came around very excited saying, "Don't you realize there's a danger of AIDS?", the orderly said, "Don't worry, I'm wearing a condom -- we're protected!" [laughter] I think that there's kind of a similar feeling I get about some of the computer aided design tools. We seem to be getting this warm feeling that we know what we're doing, when sometimes we don't. So I've prepared a couple of charts here more on the nature of being provocative; a philosophy session, I suppose, than a technical one. This CHART (#1) lists some things made up before I realized that a lot of the discussion was going to be computer aided design. There are just two charts, one is about what I view as some barriers. After listening the last few days I think there are some more barriers as well. Then I'll have some comments about what we can do -- industry, university, and government -- and you'll forgive me, Sol, if I speak for the other two as well, without having their opinions.

First of all, and this has been born out with a lot of things that happened in the last few days, is the compartmentalization of knowledge. This has been true for a long time but I think it's getting to be more so. I think we each don't seem to be speaking the same language. The software people don't seem to really understand the detailed communication systems problems, and sometimes vice versa; as a consequence, we have situations which I think one speaker said: "You need an expert to operate an expert system." I think this is a major problem if we're going to do really advanced communication systems work.

The second item has been very clear to me for a long time, and that's the lack of understanding of the problem. Very often we work on the problems we know how to do rather than the problems that we confront. A good example from my own experience, occurs in our attempt to analyze

and design Data Networks. There's a great deal of emphasis both in the analysis and in the simulation of networks on issues like throughput and delay. Queueing theory is used extensively and if you look at the analytical queueing curves they all look similar. In fact I have a queueing theorist friend of mine who says all queueing curves look identical, it's just a change of scale. [laughter] Furthermore, the things that have been interesting to me in these curves, namely, beyond the knee, are the places where I am told by people that actually work networks are never interesting. You never even want to get close to the knee of a curve. In fact, most of the delays in real networks are not due to queues. They are more often due to protocol processing. That was an example, I'm sure there are many other situations like that where we focus in on things which we think we understand, but the real problems are elsewhere. I think this idea came up in discussions of other issues as well as including those involving computer-aided design.

The third point is integrating the old with the new. I think there were several good points made yesterday by the last speaker from Fort Monmouth (Randy Reitmeyer) who indicated that most of the cost of equipment and the problems they have comes at the back end, and we spent so much time on development, which is the front end. Well, that may be true in DoD but it's not been true in the telephone industry. The telephone industry designs things to last for forty years, they always did; maybe now the time frame is 20 years. More importantly they are really effective on doing things which implement a system which allows interoperability with old things as well as the new. I think that takes a mind set which very often is not present in the people who do advanced systems for DoD. For example, I'm sure that if you go within a hundred miles of here you'll find crank telephones, they still have them in the United States. They have to interoperate with the latest digital switch that's been introduced in the last week and they do.

The fourth point is "Gresham's Law" as translated to goals. "Gresham's Law" is a law in economics that says the bad money drives out the good and the same thing is true with short term vs. long term goals. Very *often*, (and this is especially true in industry and sometimes in government as well), the short term goals are so dominant that we lose sight of long term objectives. In my opinion the word "advanced" implies a long term goal, and I think that's an important issue.

Finally, and I think this is the most important barrier, there are *insufficient* numbers of wise, articulate, and persuasive people who lead the way. I'm sorry to say I don't think I'm one of them, but I think there are enough people in this group that are in that category. But even if everybody in this group were in that category, I don't think it would be enough because in order *to do some of the things that we need to do* for advanced communication systems engineering we will need people of that calibre more than we will need computer aided design tools. We'll need both, of course.

So the question is what to do and I have a number of suggestions. Again, many of them are in the category of motherhood, but nevertheless I feel they have potential. (CHART #2) For universities I think there's an important issue of focusing on critical mass research projects. I think things are becoming far too complicated in

communications systems engineering to allow small efforts to be successful. Unfortunately universities are very often hamstrung by not having either the equipment or the critical masses of people in a particular area in any department that actually can do things that are going to make an impact. One possibility however is to implement feasibility projects for novel ideas. Of course such projects have to be funded; I'll get to that later. The third suggestion for universities is to give students hands-on experience in problem solving and in CAD models. In fact I was very pleased with the offerings of both BOSS and the Motorola system, and I personally intend to get them because I think it's very important for students to have the ability to try to do things which are real and not just Mickey Mouse things that can be done analytically. And, finally, I suggest collaboration with industry. I, myself, and many of my students have been supported through industry rather than government for the last few years. We have a rather large activity in an Industry-Liason program with Washington area industry groups, and that's been very productive. They seem to be happy and so do we.

I think the big problem I can see -- in fact particularly in industry groups I've worked with -- is that they're very mercurial. They say they have long term goals but they don't really. They assert the fact of long term research and then a year later they change their focus somewhere else, and you just can't maintain momentum that way. The second thought about industry is to be receptive to novel high-risk, high-payoff ideas from whatever source whether it be from universities or internally within an industry. Some industry organizations are very good at that. They usually are fairly large organizations. But the moderate size ones are looking at the bottom line, and the bean count is very often dominant in determining decisions about long term, especially high-risk, ideas. And the third very important issue for industry is not only to assert long term objectives but actually to commit the manpower towards those ends. The National Science Foundation, for example, has an encouraging project to try to have collaborative industry-university activities, and one of the things they insist on now, having learned their lesson from past experience, is that if industry cooperates with a university on a collaborative project, it should assign people permanently to that project until its completion without being sidetracked for proposal writing, putting out fires, and other things like that. Finally, industry ought to collaborate with universities. It's a good thing for both universities and industries.

Now, what can government do? I don't think government should play the role of simply footing the bill. First, they do have a very important role to play in coordinating and encouraging new ideas and constituencies, and I think this workshop is a very good example of that. Second, the government should increase funding cycles. It's kind of absurd to have advanced ideas which take a fairly long time to develop let alone to put together, with a funding cycle that goes from year to year. In fact, you never even know at the end of the budget cycle whether you still have the budget. For example in the telephone industry which I used as an example previously, very often budget cycles are ten years. They'll go into a big project for designing a new kind of a switch and it'll typically be a 5, 6, 7, even 10 year project. I think it's essential that the same be done by government. Of course we have to convince the Congress to do that, and I have no propo-

Barriers To Advanced Communications System Engineering

- o Compartmentalization of Knowledge
- o Lack of Understanding of The Problem
- o Integrating The Old With The New
- o Gresham's Law Translated To Goals
  The Short Term Drives Out The Long
- o Insufficient Number of Wise, Articulate, and Persuasive People

CHART #1

What Can We Do?

Universities

- Focus on Critical - Mass Research Projects
- Implement Feasibility Projects For Novel Ideas
- Give Students Hands-On Experience In Problem Solving, CAD Models
- Collaborate With Industry

Industry

- Encourage Long-Term Projects
- Be Receptive To Novel, High-Risk, High-Payoff Ideas From Whatever Source
- *Committ Manpower and Resources To IR&D*
- Collaborate With Universities

Government

- Coordinate and Encourage New Ideas and Constituencies
- Increase Funding Cycle
- Initiate and Fund Large Scale Innovative Projects In Collaboration With Universities and Industry

CHART #2

sals on how to do that. Finally, government should initiate and fund large-scale innovative projects in collaboration with universities and industry. The government in the United States I think has been lax. I know for example that in France and also in Canada -- I spent a sabbatical in Montreal -- there is a marvelous arrangement between government, industry and universities. In Quebec they have what are called National Institutes for Research, -- the one I was in was called the National Institute for Research in Telecommunication. The Institute does original research and, in collaboration with universities, give Ph.D. degrees. No courses, but they can award a degree, and it's part of an arrangement with several of the major universities in Canada. Almost all of the funding, it turns out, comes from industry, about 20% of it comes from the Province and about 10% from the federal government. But in the Telecommunications Institute, almost all of the industry funding came from Northern Telecom. I think that's a role model that perhaps the U.S. government ought to look at. Again this requires an articulate and persuasive individual, maybe not one of us but perhaps some articulate and persuasive congressman who also happens to be an engineer who understands the problem.

But that's basically what I have to say, and I'll be happy to turn it over ....

GOLOMB: Thank you. I think next I'll ask Dick Booton to make some comments.

BOOTON: I have a few charts here to serve as background for what I'm going to say. I'm not sure I believe everything that's on the charts [laughter] but I want to say a few words about the government role and the industry role and the university role; a little bit about all three. Of course I've mostly seen the industry side of things, but my comments will be on how the three ought to work together. Primarily interpreting this in terms of what these three groups should do to advance the state-of-the-art in system engineering and its practice.

One thing of course, it may sound silly, is to recognize that there is such a thing as system engineering. I'm not always sure about that myself, that there really is something called system engineering. I ran a workshop at NAECON last week on aerospace system engineering. One of the major questions was, is there such a thing, and should it be recognized as a separate discipline, should there be conferences on it and so on, or is it just a miscellaneous exercise that is part of building communication systems and airplanes and so forth? I think the government is beginning to recognize this, especially on major programs, and is willing to pay for it on funded programs.

(SLIDE 2) Government can do a lot to encourage and allow industrial support of university activities. These first two comments pertain only to government contractors, but allowing companies and encouraging them to fund research through the IR&D program is a major thing that has happened largely in the last four years. In allowing companies to support industrial affiliates and consortia through overhead this support has been counted as valid operating costs. One of the major things that the government has done in the last two years in many areas of engineering is to organize and at least get beginning funding to Centers of Excellence. Both the National Science Foundation and DoD have done this -- and I want to talk about this at the end as it pertains to com-system engineering.

(SLIDE 3) Commercial industries have a different approach. Again I have to recognize that there is such a thing as system engineering. That wasn't always the case, I know, in that it was always recognized in building a communications system that somebody had to calculate an RF budget and make sure that the dB's are all added up. But the configuration of the system was quite often done by hardware designers and things were sort of stuck together, and it really has developed over the last twenty years, that art of configuring major systems. Of course that's something that's been done for a long time in the telephone industry, for example. But as far as major com-systems of the type that we've worked on, satellite systems in particular, this has grown. I think it's recognized now as a very important thing. And again adequate system engineering tools I think is something that has grown very recently. I think all the initial efforts on CAD were devoted towards circuit design and then a little bit of tying into how do you layout a PC board. It's only recently that things have gotten beyond pen and paper, and almost the slide-rule, because I'm so old, pocket calculators and so on, and occasionally use of a large main frame. I think the advent of the PC and of modern workstations has done a lot to make these tools possible. One thing industry can do and should do is to work with universities and government to advance system engineering, and I want to come back to that in a minute. Industry can help to contribute to a more general understanding within the engineering profession of what system engineering is. The workshop last week where I was talking at, the Aerospace and Electronics Systems Society of the IEEE is devoted to taking on as its challenge to try and recognize system engineering and worry about how it should be fostered.

(SLIDE 4) I've got a chart of universities -- it's kind of curious because it's a very short chart and that wasn't intentional. I think I got tired of making these up, [laughter] but in fact one of the questions is: What should universities do? The classical thing that's been done by universities is in the development of analytical tools. All the things associated with modulation theory, noise theory, spread-spectrum theory, and so on. Those are usually efforts associated with strong individuals. I think one of the comments that the previous speaker made that I can agree with, is that universities are not used to working on large programs and in large groups. In fact, one of the problems I've seen at many universities is the fact that if the faculty is made up of a number of strong individuals, they like to be individuals, and they like to ask for their own funding and their own research projects, and they don't like to work together. There are two or three exceptions I can think of, but the general tendency is not to work on large efforts, and that gets in the way of trying to organize any of these Centers of Excellence. Things like fundamental software tools is something which has been growing, and I'll talk about BOSS in a minute. The general problem of education, that's just one line, but of course that's one of the major things that government and industry see as the job of the university. I apologize for having a short list.

(SLIDE 5) There are a number of examples of companies working with a university on a specific goal. I've been talking about BOSS some other times this week but I'd like to talk about it one more time if you can bear with me because I want to explain the background. We had a program that we used for years for simula-

tion of communications systems called LINKSIM, and that was developed probably 10, 12 years ago, and it was beginning to be sort of obsolete, it wasn't very flexible. It really was a simulation program and nothing more. We had conversations with Sam [Shanmugan] at Kansas because Sam's been very active in the development of simulation tools, and then we funded some work at Kansas which finally led to this program called BOSS. I should explain what BOSS is. I think in all the conversations this week, that no one has really explained what it is. It is not a simulation program. It's an environment to build a simulation program. That's why it's a fairly complex program, and it's real intent is to allow the engineer to, by drawing a block diagram, build a simulation program without having to write any code. In fact, BOSS writes Fortran codes. So it's kind of a supersoftware system or in a way it's an operating system. We started out funding this work thinking that we were going to build something like LINK. I should explain that most companies in the kind of business we're in have some kind of simulations program, especially for satellite links. Hughes has had one for years, we've had one, and other companies have too, and these were always thought of as being proprietary tools. In fact we guarded these very carefully. We had a big fight on this subject with NASA since they wanted a copy, and we wouldn't give them one because that was one of our proprietary tools. So that was the frame of mind we had when we started working on this new version of LINKSIM, and once it was almost finished we realized we had a problem because we had planned to take this and lock it up at TRW and we had paid for it so it belonged to us. Sam thought that this was something very general and very important, and ought to be made available. We had several very heated meetings and we finally decided one thing that the core of the system, the core operating system which we call BOSS, should be made available in general. In fact we arranged that Sam should do that for universities, and it's been advertised in the *Com Society Magazine* and so on. To make a complete tool out of this you still need to add to this special application programs so that our thoughts are that some of the specific programs that do have some proprietary interests we'll keep separate. But the basic system will be public.

I think that BOSS was, from our point of view, an outstanding example of a university-industry joint development. It's true though that there aren't many like this, and I think that one of the things that make this stand out so well is that most of our efforts haven't gone this well in terms of producing something which was really a joint development. First of all, I think it's a real challenge to find efforts and to find professors, and to find people at the companies that can work together closely. I can always bring this up as a great example of university-industry working together, but I'm always embarrassed that it's the only one I can think of. So I think that there is a real challenge in how do you do more of this, and produce not just another master's thesis or a doctoral dissertation that gets filed away, but something that's really useful.

(SLIDE 6) One last comment gets back to the Centers of Excellence question. I think it's possible to have a Center of Excellence for Advanced Communication System Engineering and I think of this in terms of the kinds of things the National Science Foundation and DoD can do. There are several problems in doing this.

TRW

Role of Government, Industry, and Universities in Advanced Communication Systems Engineering

Richard C. Booton, Jr.

29 May 1987

SLIDE #1

TRW

Government Role

- Recognize systems engineering as a discipline and as a vital task on programs
- Encourage (and allow) industrial support of university activities
  - Directed research funded through IR&D
  - Industrial affiliate (including consortia) through overhead
  - Tax incentives for computer and other equipment contributions
- Participate in organization and funding of centers of excellence
  - NSF and DoD funding of engineering research centers and similar centers

SLIDE #2

36607 587

TRW

Industry Role

- Recognize that systems engineering is a vital discipline and an important task on any communication systems development
- Give systems engineers adequate tools and opportunity to use them
- Develop training programs
- Work with universities and government to advance the practice of systems engineering
- Contribute to a more general understanding within the profession of what systems engineering is (workshop at NAECON '87)

SLIDE #3

46608 587

Universities' Role

TRW

- Develop analytical tools (modulation and noise theory)
- Develop fundamental software tools
- Educate students in communication systems engineering

SLIDE #4

86609 587

Example of a Joint Industry-University Activity

- TRW needed to upgrade its communication link simulation (LINKSIM) (also known as LINK)
- Kansas University (KU) had been active in development of simulation (Sam Shanmugan)
- TRW began its own developments and also funded research at KU
- KU concentrated on the core software (BOSS)
- Following successful completion of the joint development, the merged results (known as LINKSIM II) were used on Space Station contract
- KU is making the core software (boss) available on a more general basis

SLIDE #5

365010 587

TRW

Possible "Center of Excellence"

- A center of excellence for advanced communication systems engineering is possible
- Both government and industry must believe in the seriousness of the need
- Industry has to believe that advantages of working together outweigh disadvantages (protection of proprietary data)
- Recent examples
  - UCLA ERC in hazardous waste
  - U of Colorado ERC in optical computing
  - U of Colorado UIRC in microwave/millimeter wave CAD

SLIDE #6

One is that both government and industry don't believe that this is worthwhile. I think most of the people in this room, because you came here, think that com-system engineering is important. I don't think very many people in the government recognize that this is an important need; there are all kinds of important research needs, but I don't think it is recognized that this is something worth spending time and money on. That's the biggest barrier I think to getting either the National Science Foundation or DoD behind an effort like this. In it you have to believe that not only it's a good thing, i.e., that the results will be valuable, but that it's worth the disadvantages of revealing some of your or some of the company's proprietary data. I've picked out three examples from last year centers -- these are all National Science Foundation efforts -- but recently UCLA got an effort in an Engineering Research Center from the National Science Foundation in hazardous waste, and one comment on that is that I may insult some very nice people but I don't think that that's very high class research. I think it sold because hazardous waste is recognized as a very important problem and that it's well worth spending money on. So I think that's an example of a very important problem, but the problem is so important that it would be funded.

There's kind of an opposite one at Colorado. They just won an ERC in optical computing, and I think that is a very long range, very risky effort, but it's very high class research. So occasionally things are funded because they're high class research, but are very long range. They are very interesting things that might someday be very, very important; but are not going to be in the near term. There's a third one which K.C. Gupta will pardon my talking about -- and I got the letters wrong -- is IUCR, Industry University Cooperative Research Center, which at Colorado is in microwave and millimeter wave. This is an interesting effort because it's not near term, but it's not very, very long range either; it's in between. It's one where a number of companies have already agreed that it's a worthwhile effort to them and they've already argued through the worries about proprietary data and so on, and are willing to support that. I think that's a very good example of the kind of middle of the road between a very, very short range important problem and long range research. If you look at all this in terms of communication system engineering I think there are some real problems in trying to first define a problem that both government and a variety of industries will think is important, and important enough for the companies not to worry about their own proprietary data. I think that if those two things can be done, I think there's a lot of room for a very strong center. But I think it's a very hard battle to win because even though we all think com-system engineering is very important, I don't think the average man on the street does, or the average man in the National Science Foundation or DoD. So that's a challenge -- I'd appreciate any comments on that especially.

GOLOMB: Okay, thank you Dick. I now will ask Bill Sander to make some comments ....

SANDER: Some of the material that I'm going to talk about today has been prepared for some time (VIEWGRAPH #1); other material I prepared on Monday and then found out that the guy in our office that owns the Macintosh computer took it home with him over the weekend -- so I didn't get to make good slides. Other material I just fixed up about an hour ago.

One of the key questions I guess that everybody is interested in is how are we going to get money to do all these things, and how do you keep the microphone from falling off [laughter] .... There, let it rest on the buttonhole. The prognosis is not very good and of course we've heard two speakers this morning talk about Centers of Excellence, or the block funding concept. The problem with that is it creates a situation that's great for very few people, but very bad for a lot of people, and that is that the history of implementation of that kind of thing has been zero base funding. We don't know all the reasons, I can't explain why. But my personal budget has taken a 50% reduction from last year, this year over last year. It went from about $1.7 million to less than $900K right now. That may be due to some of the block funding projects, it may be due to just a philosophy. One of the explanations I've heard are that it's the philosophy of the Army staff in Washington that basic research should not be done by the Army in their laboratories but rather should be left to industry, and when we have some hardware that we want to obtain we simply ask industry to design it and provide it. Well, whatever the reasons, whether it's that, whether it's the Gramm-Rudman Act or whatever, funding appears to be very hard to come by, not just at ARO but also at the laboratories that are being cut at the 6.2 and even the 6.3 levels of funding. Of course ARO is all 6.1, which is basic research. There's been a lot of cutting this year. Why, I don't know, maybe some of you know better than I do; I've heard a number of possible explanations.

One thing that I feel very strongly about, and a point I want to make today, is in this business of Command, Control, and Communications (C3I) a lot of the emphasis is being placed on Command and Control (C2) and not so much emphasis on Communications. (VIEWGRAPH #2) This chart shows where communication sits in the whole picture, and it is a weak link. The commander who makes decisions can't get information from the field or out to his units that he must command if he doesn't have communications, and this is the area that's going to be attacked by the enemy because it is an Achilles heel, a weak link in any situation. But I think when that you hear about DoD programs in C3I, most of what you hear about is C2 and the Intelligence (I) part and not the communications part, and I don't see much in the funding lines, very large dollars, in the communications area. (VIEWGRAPH #3) I think these headlines in the *Washington Post* and *Jain's Defense Weekly* emphasize what I've been saying. There's no doubt, it's in the public news media, that the Soviet Union's philosophy is to attack the communications, and that's very clear. I don't believe people in Washington, D.C. are paying attention to this for some reason. I don't know why.

One thing I observed during the workshop is that there's been a lot of talk about cable systems and about satellite systems. These systems are certainly important, but the Army's systems are very, very much different. (VIEWGRAPH #4) They're ground to ground systems, they're going to be very large complex networks, a thousand nodes or more, very dynamic topology and connectivity. In a little while, I'll show you what the requirement is. Army forces are supposed to move very rapidly, and in a wartime situation things are going to be knocked out, you're going to lose nodes due to fire, so it's very dynamic. We're going to be in the electronic warfare environment and because of

the emphasis on C2, the command and control, there's going to be a tremendous increase in the amount of data that has to be communicated on these networks, and this puts pressure on the network. Many of these things I haven't seen addressed today, so I go back to what I said before. Even within the Army I don't see enough emphasis on the kinds of communications problems that we need to solve to provide our commanders with reliable communications in the next war. I'd like for us all to get the word out that we need to have more emphasis in this area. It's very important to the survival of Army communications -- I did my active duty as a Signal Corps officer, and I know how Signal Corps officers in the Army think. I know how the commanders whom they serve with communications think. They want to be able to pick up a telephone, send a message, and get an answer right away; and if they don't, they get very upset. There's good reason for that, particularly in a very fast moving warfare environment, if they can't control their forces they're going to start losing. Inclusion of ECCM in networks and consideration of the very dynamic nature of the network are very important to us. Even though I don't have any money I'd like to see more work done in these areas. (VIEWGRAPH #5)

Okay, this is straight from Army TRADOC, Training and Doctrine Command. In good conditions, they want instantaneous communication with 98% reliability. One echelon of organization up, down and lateral. The units may be scattered out and noncontiguous. In an electronic warfare environment, they're willing to accept 75% reliability. Because of the dispersion and mobility of the units, we want to be able to communicate to 1,000 Km with equipment that each unit owns themselves, communicate all data while moving (this is the mobility factor), and all transmissions secure. These are some pretty hefty requirements on communications systems and I don't think we have the ability to meet them right now. I don't know when we're going to have that ability, and that concerns me. There are a lot of things we can do. This is just one example. (VIEWGRAPH #6) I put this up because it shows a lot of different things, and also because in the encoding area, you talk about a few dB's of gain, maybe even 1 dB of gain, but with adaptive antennas you can talk about 25-40 dB of gain. So maybe adaptive antennas could be very contributory to reaching the goals that we need; however, if we do have adaptive antennas in the network, we have to know how to control that network with the adaptive antennas and how to control each antenna at each node in the network, and we don't know how to do that right now. Additional information has to be distributed through the network which increases communications overhead. At this point I'd like to interject a possibility that we may need to back up and try to think a little more basically about networks. What is the minimum amount of information that you need to be distributed in the network, and how do we get the maximum performance of the network? So the first thing that comes to my mind is what do you need to know -- you need to know position of the nodes, you need to know the frequency that they're operating on, and you need to know something about codes. You need to know a pseudorandom code or an encrypting code in order to access the signal. That, in my way of thinking, is about all you need to know. But you want to minimize that information. You may be able to get position information from say a GPS satellite,

and with that position information at each node maybe you then could do something to control all of these adaptive antenna arrays at each node, to null out jammers, to null out interferers, to minimize the spatial distribution of your own transmission in order to minimize interception of that signal, and interference with your friendly neighbors, and to steer beams in the directions that you want to communicate with. I would like to see some thinking done in this area because everybody tells me we don't know how to design distributed communication networks, and when people start trying to think of how the routing tables are designed they have to start thinking of new kinds of network architectures and protocols because routing tables get too big and cumbersome and you start increasing the delay in the network just to get messages through it. What the Army commanders are telling us is that they want instantaneous communication. I don't know how we're going to reach that.

Okay, we're changing directions a little bit to discuss some things that have already been mentioned. (VIEWGRAPH #7) I say "think big". I am not sure that thinking big is the best way to approach this problem. I do believe that a Center of Excellence or even two Centers of Excellence would be important. I think there are a lot of things that have to be accomplished in such an environment. There must be interdisciplinary interaction with many people talking to one another and exchanging ideas. Right now the philosophy of funding is big, and you know many examples of these: the SDI Program, the University Research Initiative Program, the University Research Instrumentation Program. We have fenced money, about $2 million a year for AI, in our division. The Ultra-Submicron Electronics Research Program has been fenced, and you know about the VHSIC program which is not truly basic research. These are big programs that are block funded, a few examples. As I said before, there are some disadvantages to doing this. You do have the advantages of symbiosis in interdisciplinary fields and you share expensive facilities. I don't know what kind of expensive facilities are required in the communications area other than computing equipment. The disadvantages are the little guy loses. It's very difficult today for me to come up with sufficient money to get new university professors started in basic research. They will have to tag along with a better known researcher in the university in order to get started today. You lose a lot of creativity from the individual thinker when you can't fund the small guy. I think one of the reasons for the Small Business Innovative Research program in the government is stated to be *that most innovation* comes out of small business, the real creativity comes out of small business. So we should encourage that. Well, we're not encouraging that on the other hand in the university part of the program. And, as I said earlier, it comes out of your hide, it's zero base right now. You end up performing research by committee and the old picture of the swing going through the tree comes to mind every time I think about that. Also the rich get richer, the big institutions with well established research programs simply get richer, the opportunity for other institutions to build their programs is minimum, and you lose flexibility. (VIEWGRAPH #8)

Just before I came to this meeting, I was reading this document, which you ought to write off to your congressman and get; it's the April 15th, 1987, House Armed Services Committee Report on the Authori-

zation Bill. You can get a lot of insight into what is going on and how to get money just by reading the way Congress thinks. One of the things they said in that report is that $1 of research today costs the government tens of dollars in a 5-10 year period because we've come up with great new ideas that we have to procure equipment with new performance. So that's a philosophy, that's the way they're thinking up there today. They have totally wiped out, or nearly totally wiped out, you can see the reduction here in numbers, the LHX Helicopter Program. They don't agree with it, they won't buy it. I tend to agree with Congress on that; I can't imagine a helicopter with one man in it that has to fight in a war. Services have too many programs that are nice to have and are duplication of effort. They have increased the authorization for what we call the Conventional Defense Initiative (CDI), and this is politics coming into the picture. SDI is Reagan's program, CDI is a congressional democratic party kind of program. Congress is increasing CDI because democrats have control. They also are increasing the URI program, University Research Initiative program. Last year each service had a line in their budget for URI funds and this goes back to the zero base funding problem. Congress found out, there's a whole page in this report on their philosophy that the services simply gave the URI money to ARO and didn't increase their budget. They included the URI money as part of their budget, so in other words we had to pay the URI bill out of our hide. Congress found out and didn't like that; they're very much in favor of funding university research. This year there is one line. They took all the lines out of the services' budgets and put one line in the DoD budget. They're hoping to eliminate the problem.

These are some of the specific paragraphs from the National Defense Authorization Act for FY 88/89:

(1) Computer microcircuit testing is a program to provide the DoD special kinds of integrated circuits that are needed which you can't buy from industry, small quantity lots and things of that nature.

(2) X-ray lithography they thought was important. This is a project that funds DARPA to investigate optical processing.

(3) Then we get into electronic warfare, and this is the part that's really important to all of us. In 1985, Congress asked for a master plan from DoD; DoD said they couldn't provide it because the services were not cooperative. Congress fully believes that this program is not coordinated and has a lot of duplication of effort in it. They also believe that existing equipment cannot meet the threat. In other words, they believe that there is a great need here; and this is reflected in the amount of funds that they have provided in this paragraph in the House as part of the Authorization Bill. What they have said is you can't give more than $50 million to each service for this program until this plan that we've requested is in the hands of Congress for at least six weeks.

So you see the kind of battles that DoD level fights to try to get money. (VIEWGRAPH #9)

Randy Reitmeyer of ETDL pointed out earlier this concept of Nonavailable Parts, that the lifetime of an IC is only about five years, but equipment is in the field for fifteen or more years. So we have the problem of non-available parts, and how do

we get around it. I went to a meeting once in Washington and some of the proposals were stockpiling the components when you buy the equipment in the beginning, which was deemed a little unfeasible; maintain the old fabrication lines so you can continue to build the parts and pay an overhead which is not very feasible; redesign IC's -- I just got a $20 million proposal for one year from a university for an Engineering Center associated with a university to build a facility to design replacement IC's for non-available parts. I believe that Congress is very much involved and it's going to end up being the same as a university set-aside such as occurred during the URI program initiation last year. In other words, it probably will be funded. I tend to disagree with the concept of redesigning IC's. I agree with the concept that Randy presented where you redesign the circuit boards and install the new boards using existing, available parts. (VIEWGRAPH #10)

While we're on this subject I'd like to address some of the issues of CAD. ARO has been supporting for over a decade a CAD program in integrated circuits. I see much similarity, many parallel lines, to what we are addressing in this workshop. I didn't realize how much was going on in CAD of communication systems until I heard the things that have been discussed here. CAD of IC's went through similar things over the last fifteen years, from very detailed low level kinds of CAD tools to higher level tools. Today, there is much emphasis in CAD of IC's on trying to couple different levels of simulation. I think that's an essential part of CAD, the interfacing between the different levels of design and identification of the parameters to use to interface between the levels. When you complete design at one level you've got to be able to transition to the next level of design, whether it's link design and then networks or vice versa. This is a problem that the CAD people with integrated circuits are running into today as well. They're also trying to put algorithms on parallel computers, which has been discussed here, and also using hardware accelerators. I do believe that the integration of levels of design is an important part, particularly for the problems that the Army has where we have large system designs. (VIEWGRAPH #11)

One of the messages I hope I gave you in talking about some of the Congressional information from the House Armed Services Committee Bill is that Congress expects funds that are spent by DoD, or the Army, to be addressing DoD and Army unique problems. I think we certainly can define Army communications as being unique, not only from public communications but from the Navy and the Air Force as well. My message to you is to think about relevance and uniqueness when you write proposals. Get the message on the impact of block funding government. The Defense Science Board and the Army Science Board have critical input to Congress. The report from the House Armed Services Committee points out that Congress can identify specific needs and say you will do this, because it is a need even though you didn't identify it to us. The Army Science Board has critical input to establishing paragraphs in these bills which appropriate large amounts of money for specific programs. Professional associations such as the Electronic Warfare Association and Defense Science and Electronics also have an impact. If you have an opportunity, write an article for one of these magazines or trade journals; because the higher level DoD staff in Washington and at the Pentagon read them. They also attend meetings

UNITED STATES ARMY
RESEARCH OFFICE

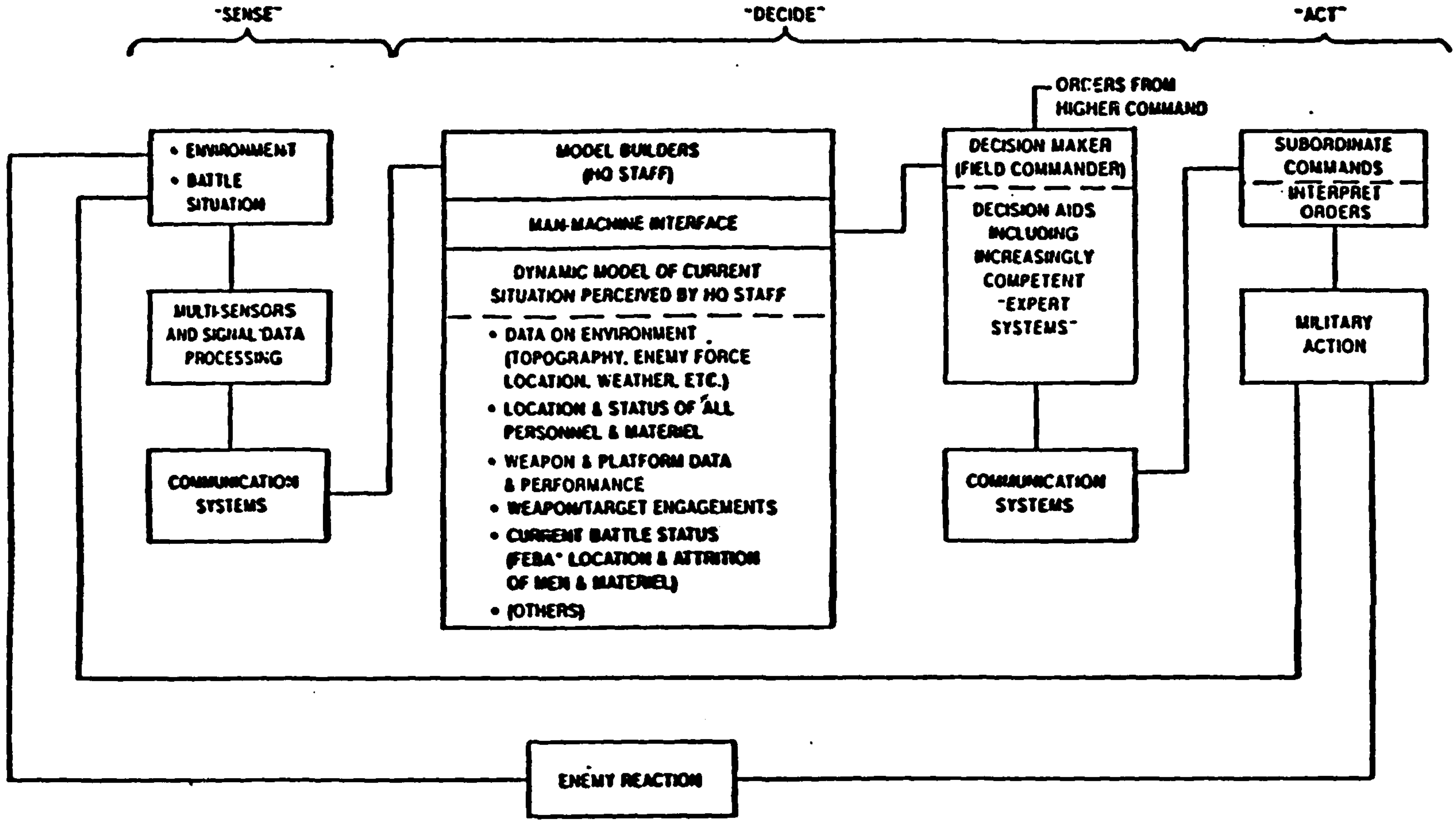
BROAD AGENCY ANNOUNCEMENT

P.O. Box 12211
Research Triangle Park, N.C.
27709-2211

May 1986

VIEWGRAPH #1

C^3I



*FORWARD EDGE OF THE BATTLE AREA

Simplified representation of the C^3 process.

VIEWGRAPH #2

ARO ELECTRONICS DIVISION

ARMY COMMUNICATIONS

"Jane's says Soviets make NATO jamming top priority"

"Soviets consider disruption of NATO communications a top priority"

Washington Post, June 11, 1986

"The military implications of inadequate communications"

"Data communications is the crucial aspect of C3I - without it the rest could not function."

JANE'S DEFENSE WEEKLY
22 MARCH 1986

VIEWGRAPH #3

ARO ELECTRONICS DIVISION

ARMY COMMUNICATIONS

FUTURE ARMY COMMUNICATIONS WILL BE DIGITAL SPREAD SPECTRUM PACKET RADIO, FIBER OPTIC CABLE, AND MILLIMETER RADIO.

0 CHARACTERISTICS

> LARGE, COMPLEX NETWORKS
> VERY DYNAMIC TOPOLOGY & CONNECTIVITY
> ELECTRONIC WARFARE
  o ECM
  o ECCM
> RAPIDLY INCREASING DEMAND DUE TO C2 EMPHASIS

GREAT EMPHASIS ON C2 NOT ENOUGH EMPHASIS ON COMMUNICATIONS

VIEWGRAPH #4

ARO ELECTRONICS DIVISION

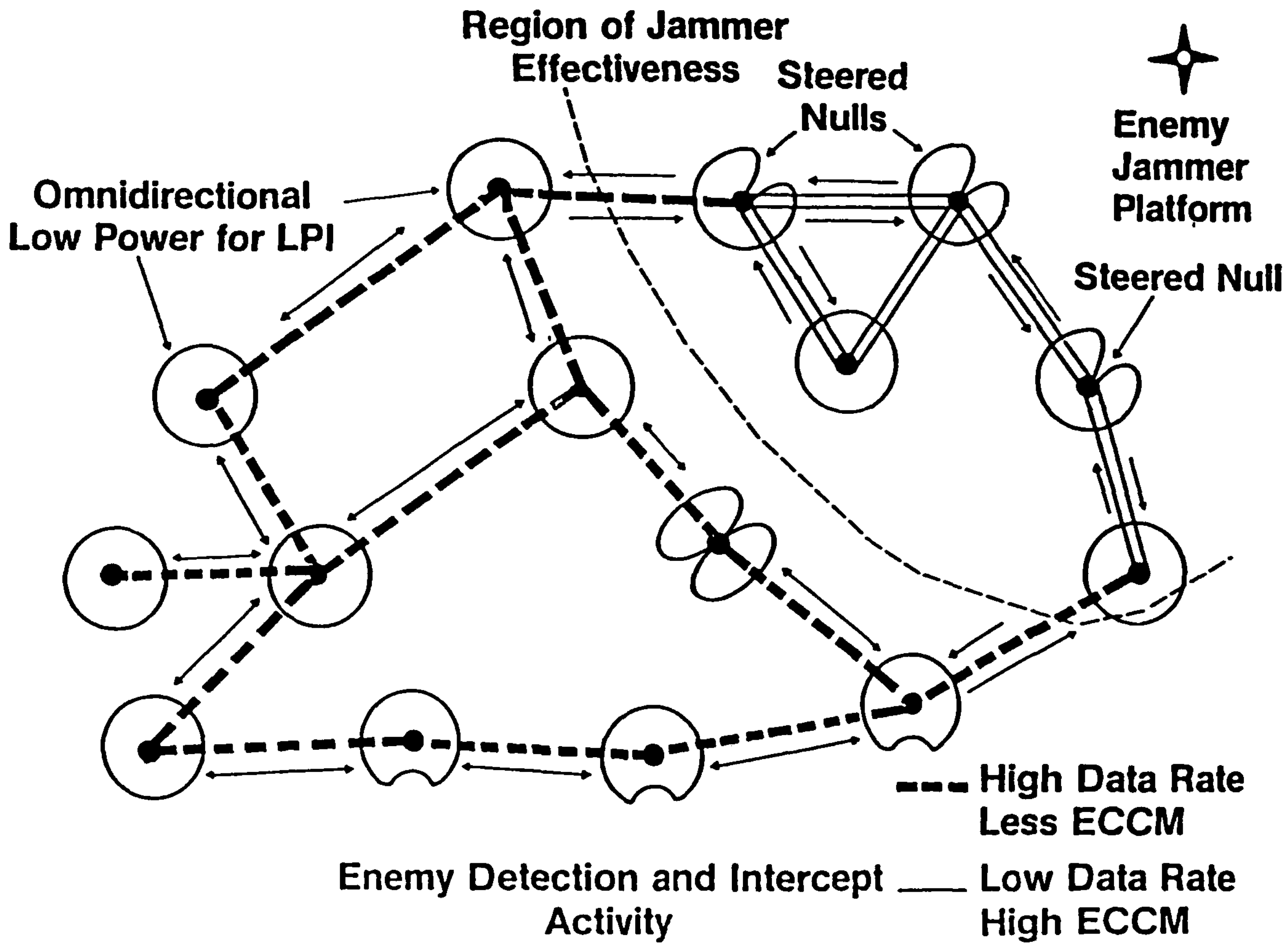
ARMY 21 COMMUNICATIONS

- Instantaneous with 98% reliability one echelon up, down, and laterally.
- In active countermeasure/EMP environment, communications not less than 75% reliable.
- Tactical units communicate to 1000KM with organic means (unmanned relays).
- Communicate all data while moving.
- All transmissions secure.

VIEWGRAPH #5

E43

ADAPTIVE ECCM NETWORK



VIEWGRAPH #6

FUNDING COMMUNICATIONS RESEARCH

- O THINK **BIG!**

  Congress, DoD, DA, DSB, ASB, ADPA, Defense Electronics, DS&E, Journal of Electronic Defense

- O Need Paragraph in Budget!

- O Trend is block funding.

SDI	URIP	USER	But ....
URI	AI	VHSIC	

ADVANTAGES	DISADVANTAGES
Symbiosis	Little guy loses
Interdisciplinary	Creativity loses
Share expensive facilities	Comes out of your hide
	Research by committee
	Rich get richer
	Lose flexibility(Entrenchment)

VIEWGRAPH #7

NATIONAL DEFENSE AUTHORIZATION ACT FOR FY88/89

House Armed Services Committee 15APR87

- One dollar today ——► tens of dollars in 5-10 year period
- Not fully support Army's LHX ($397M-$227M)
- Services have too many programs

  "nice to have"
  "duplication of effort"

\+ Increase authorization for CDI($328M) and URI($191M)

PARAGRAPHS:

Computer micro-circuit testing (DoD +$10M)

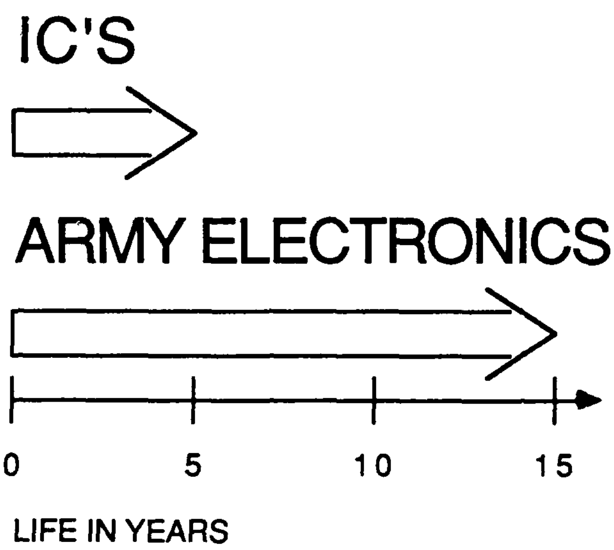
X-ray lithography
(Beyond VHSIC i.e. submicron. Not requested by DoD. +$20M)

Optical Processor Identification Technology
(only for optical DARPA +$5.5M)

Electronic Warfare (DoD +$486.8)
Not coordinated - duplication
Equipment lags threat
Request in 1985 for Master Plan
Uncooperative attitude of services
< $50M each service until plan submitted

VIEWGRAPH #8

NONAVAILABLE PARTS



APPROACHES

Stockpile components.
Maintain fabrication lines.
Redesign IC's.
Redesign PCB's with available chips.

VIEWGRAPH #9

ARO ELECTRONICS DIVISION

CAD

PROGRESS:

- O SPICE - CIRCUIT SIMULATION
- O SUPREM - PROCESS MODELING
- O PISCES - DEVICE ANALYSIS
- O SAMSON - CIRCUIT SIMULATION
- O CINNAMIN - CIRCUIT ANALYSIS
- O WASIM - WAVEFORM SIMULATION

FUTURE WORK:

- ● Coupled Process/Device Simulation.
- O Physical Models for Submicron Devices.
- ● Algorithms for Parallel Computers.
- ● Hardware Accelerators.

INTEGRATION OF LEVELS OF DESIGN

VIEWGRAPH #10

ARE ARMY (DoD) REQUIREMENTS FOR AUTOMATED DESIGN UNIQUE?

VIEWGRAPH #11

AUTOMATED DESIGN OF COMMUNICATIONS REASON TO BE

- O No closed form solution to large scale communications problems.
- O Non-available parts.
- O Avoid technological surprise. (In-process design for new technology)
- O Time and cost to design.
- O Need simulation and modeling.
- O Performance evaluation and testing.

VIEWGRAPH #12

sponsored by associations such as the American Defense Preparedness Agency. You can sell them on the need for communications research, and that's the job we all need to do right now. Communications research needs a block of money set aside because that is the philosophy of management in Washington today. The funding strategy is to set aside in big, major programs, blocks of money, to do things that are needed. I truly believe that one of the biggest needs right now is communication system design work, especially for the Army. It may not be as critical to the Navy and the Air Force, but for the Army I think that it is extremely critical to the survival of our forces in a war.

I think CAD is very important. Automated design tools will be very important because the large, complex networks that in the Army environment cannot be designed without *computer-aided*, automated design tools, not completely enough to satisfy the need and come up with the best designs. (VIEWGRAPH #12) So what are some of the reasons to be that you can cite in any articles or any appeals to the Defense Science Board, Army Science Board members or, for that matter, directly to Army staff and DoD staff in Washington. This is a list that I came up with, I don't claim that it's complete. But, these are some of the things that you see cited and that you can put down in your proposals to me or any appeals that you may make to the government part of this three-member organization that we're trying to get to work together -- industry, university and government.

GOLOMB: Thank you, Bill. I thought I would exercise the right, or the Chairman's prerogative, by making a few remarks of my own. First of all I have a tip for all of you: A proposal in the current climate of what's important in Washington that's guaranteed to sell, and the title of the proposal is, "Using Superconducting Neural Networks to Find the Cure for AIDS," [laughter] has all the right buzz words in it. One of the things I think that our panel and the discussion this morning should address is, "Why advanced communication system engineering and design tools?" to facilitate it and meet urgent national needs. I think Bill Sander helped us quite a bit in telling us what some of the military needs are, and I think on the civilian side there is a real issue involving technological superiority that the U.S. historically has had in the communications area, that we're probably losing the clear edge that we used to have. There are many large foreign communication companies -- Phillips and Siemens in Europe, Fujitsu, Mitsubishi, Matsushita in Japan -- and certainly a part of being competitive with those companies and the countries they represent would be addressed by having a superior design capability along the lines that has been discussed this week at the meeting. I think there's also a question of how advanced is advanced communication system engineering, what should we be focusing on, and, as was pointed out this morning, a lot of the discussion this week has really been simply applying the current state-of-the-art in CAD to communication systems. Obviously looking to the future we should be looking at something a bit more ambitious than that. Looking at helping the engineer-designer at the top-level system-block-diagram level where given the performance specifications and technological constraints, in a just slightly futuristic world, the computer design system would do the top level block diagram, it would select the modulation and the coding and all of these other things. This also bears on the issue of maintaining a technological

edge in that it would upgrade the equivalent education level of the design engineer from someone who was less educated to the equivalent of someone who was at the forefront of the profession. That's another area where we're falling behind. We're underproducing trained engineers in this country. I'm sure you've all heard the statistic that Japan, with half the population of the USA, graduates more engineers than the United States, and probably in Japan the engineering profession tends to attract many of the very brightest students to a significantly larger extent than is true in the United States. One thing I might mention that is likely to happen at USC ... we've been talking to Dr. Eberhart Rechtin who is currently President of Aerospace Corp. He is scheduled to retire from that position in less than a year. He is interested in starting a program in training system architects, and I think he's looking at aerospace architects, communication architects, and perhaps civil engineering architects as the three focal areas of starting a very advanced educational program, and we certainly see that as something that could interact significantly with our efforts toward more advanced system design tools in communications. I'd like to go back first to the panel itself and then after that I'll open it up to general discussion. I think I may have given Ray a little less time than the other two, and you might have additional things you want to say ....

PICKHOLTZ: Well, I agree with you, Sol, I think it would be nice to have engineers trained in a way that would elevate them professionally. We had a discussion at one of the meals, in fact it was last night I believe, about whether engineers are really professionals, and I think one of the issues that was raised is that professionals have clients and patients and engineers have employers, but that's not the issue I think you're referring to. I think you're referring to the quality of the educational level that they will have so that they would be in a position to do even more creative work than they have been doing up till now. I alluded to that in my first comments when I indicated the barriers I felt were still evident in doing advanced communication systems engineering, and that's the compartmentalization of knowledge. Engineers nowadays, even in an electrical engineering department, the electrophysicists hardly ever talk to the communications engineers, let alone the computer scientists. In fact, sometimes the various disciplines are doing the same things but using different languages. I think Irv Reed pointed out the other day that we need some kind of a hardware specification language which not only could talk to humans but could talk to computers. I think there's an even more fundamental problem and that is this compartmentalization has resulted in people who get to know, the proverbial "more and more about less and less until they know everything about practically nothing." I don't personally have an easy solution to this because I've been very disappointed myself in some of the attempts that have been made at the universities I've been at of attempting interdisciplinary programs. They tend to become very wishy washy and young faculty members, in particular, are advised to steer away from some of these wishy washy things because they don't generally lead to really deep things which can be published and be recognized by their peers. Unfortunately the peer structure is becoming more and more restrictive and the very word system implies that we have the broader picture.

I'm sort of torn between this attempt to introduce in the universities programs which transcend a lot of borders, but at the same time I lament the fact that we have students now who really don't understand what a Fourier transform is. They can use an FFT algorithm and in fact even program the FFT algorithm without understanding the implications of what it is they're doing or even why they're doing it. I think there's a fundamental problem here in the transfer of basic knowledge or what I might even call wisdom from one generation to the next. We very often re-invent things that have been done by previous generations and think that they're brand new. We very often don't learn from past experiences and that's a big concern of mine as an academic. I don't know a simple answer to the problem. I'd like to see the students take more mathematics but at the same time there are so many other things for them to do that there is a difficulty.

The only other thing I would like to point out in the business of systems engineering, especially advanced communication systems engineering, is that I'm not at all quite sure what it means. The reason I'm not sure is because things that one may not think of as being important now may turn out to be very important ten years from now. In the late 19th century, the director of the patent office decided that they ought to close the patent office because everything that needs to be done had already been invented. At every conference I go to and this meeting is no exception, there are a number of people who'll have these time diagrams of the growth of number of gates per chip versus time, and around 1990 they level off. There have probably been curves like that for everything including propeller driven airplanes. When those curves are drawn and the projections made, it's very difficult to anticipate a technology that is going to completely displace the one being discussed, and that will certainly happen. There are technologies that are just on the verge of development now, maybe it's superconducting neural networks, I don't know. There are probably ideas which are just germinating now which will suddenly change our very outlook of what's important. I'll give you one example. I think it's pertinent to the systems engineering problem. For a long time, many of the communications CAD tools I've seen have been focused on the problems of transmission problems due to limitations on bandwidth. For some technologies that are coming out now, and most notably fiber optics, bandwidth is not the problem. In fact, bandwidth is so far from the problem that the issue is how do you use the bandwidth? How do you exploit the bandwidth? How do you waste the bandwidth, if you will, in order to take advantage of the requirements that you need? Bob Gagliardi made a couple of hints along those lines of using the bandwidth of a fiber optic as a switch. He proposed the use of a spread spectrum CDMA switch for example. Other people have been proposing similar kinds of things because you can trade off bandwidth for a lot of other requirements if you're clever about it. But the cleverness is not going to come out of an existing computer aided design tool. I think it's going to come out of, as I said before, bright people who understand the broad scope of the problem and who have the ability to be creative and inventive. In my experience I've not found a way of teaching people how to invent. I don't think that's the kind of thing you can learn in school but there are environments that you can create which will stimulate invention, and that's an environment we

need in schools and industry and elsewhere. We ought to look at the problem in a larger perspective than just what the technology seems to be right now and what the issues seem to be right now. We can talk about these blue sky things, but I think from the point of view of training student in universities, it is essential that we avoid the rigid thinking that occurs when you look at everything from a short term perspective.

BOOTON: Okay, I'd like to make a comment on one of your first points, Sol. We did hear this morning about some important Army communication problems and I believe that there are many other important government communication problems. No one has ever made a good case for the fact that better communications systems engineering might help solve those problems. In fact I don't know of many people who believe it outside the circle of communication systems engineers, who really believe that we would have better communication systems if we had better system engineering. I'd be very interested in getting comments from the audience because we all sort of believe it but that may be a self-serving belief. Is there any way to get some examples and to make some arguments, so I'll leave that as a question.

GOLOMB: Okay, rather than trying to answer that one by myself I'll leave that open for the discussion period, and I'll ask Bill if you'd like to make any more comments at this point ....

SANDER: Sometimes I feel so caught up in this communications problem of the Army that I could talk probably for a whole day, 24 hours, or something like that, but I'd probably end up repeating a lot of what I'd said, and I may do that this time. We're talking about compartmented education, compartmented engineering people. One of the things I remember is when I came through undergraduate school in the early '60's, we all got a very broad education. We had to be educated in everything from solid state to electric machinery. By the time I got into graduate school, people were already starting to compartmentalize their educations, either a solid state scientist or a controls scientist or a communication/information theory kind of person. We've tended to maintain that since the days of my graduate school. Recently though I've seen a great trend nationwide in more symbiotic relationships between different disciplines, and you see that in CAD and IC's, you're having people who know numerical science getting involved with the computer aided design of integrated circuits, and I think you will find that to happen with you in the communications arena as well and other disciplines. Chemists are getting involved with the microelectronics people; you're finding psychologists and physiologists getting involved with EE with neural networks. One of the things that impresses me about CAD is that it gives you a backbone about which to build a broader knowledge of your discipline of communications. Most people who use CAD tools don't know the detailed knowledge about it. They go out and seek some more knowledge so they can better use that tool, and they broaden their knowledge of the entire field. Due to the increasing complexity both of public, civilian communications and particularly Army communications, I really believe, and I want to emphasize this again, that CAD or whatever you want to call it -- simulation, modeling, whatever -- is essential to the design of these systems. I don't believe there's a closed loop solution or analytical solutions that can be used when you have

thousands of nodes, all of which are entirely mobile.

My futuristic picture of the Army communications system is that every vehicle has an adaptive antenna on it and you don't have to stick anything in the ground to communicate. It is completely 100% mobile, you don't have to stop to communicate. You can move anywhere you want to and still communicate; everything is contained in a mobile platform; and you have an adaptive antenna that is controlled by the network. You know your position in real-time, and many of the things that you have to control in the system are based on your position and your relative position with other nodes in the network. The kind of simulation, modeling, and automated design tools that I hope will be developed in the future will be embedded in this network somewhere in processors at the nodes. We need those in order to build distributed communication networks and to control, in particular in the Army, all the complex things that we're going to have to control -- code rates, power levels, antenna directions and so forth -- in order to survive in the electronic warfare environment. So I strongly believe in this need for CAD, simulation, modeling, and automated design tools.

But as we address these things we have to keep in mind what happens to the overhead, and that's why I wanted to make the point that you have to think in terms of what is the minimum information that has to be exchanged between nodes in such networks to get the best possible performance, because we just went through the example of the packet radio overlay for SINCGARS, and the throughput of that network when the overlay was installed was greatly reduced. Now that has to do with a lot of things that were frozen in the SINCGARS design before it was even thought that it would be used in a packet radio network which made it less efficient. The reason for the decrease in the efficiency was the overhead. The problem for the Army basically is how do you manage ECCM in the networks? How do you manage heterogeneous networks, internetworking? The Army wants to go to OSI. ISO wants to enforce standards on network management in the Army, and I believe one of the reasons is they want to use a standard protocol for one thing, but the other thing is they want to be able to use existing communications -- fixed plant civilian communications wherever they are -- to the maximum extent possible; and they want to be able to plug into those things. At any rate no matter whether it's our own equipment, Army owned and developed equipment, we're going to have heterogeneous networks, we're going to have a requirement for complex internetworking. We're going to have multipriority traffic which has to be handled; we're going to have widely varying demand from practically no traffic when the war is not being fought to so much traffic the network can't handle it when the war gets really hot. And mobility, that's the last thing.

GOLOMB: I think what I'd like to do now is open this up to the audience if you have questions that you'd like to address or comments that you'd like to make .... Yes, Barney ....

REIFFEN: I'd like to offer some comments which sort of intersect with a lot of the things that people said. The theme today has been more global than perhaps earlier. The earlier themes have been computer aids whereas today we're really talking about system engineering, communication system engineering, and in a sense that's a more difficult but more provocative

theme. I think, I'll say again a comment I said on the first day which is that everybody has a clear idea of system engineering, but I suspect if we polled the room there would be a great variance in what everybody means by system engineering. Perhaps this is one of the difficulties in joining this issue because I don't think everybody's talking about the same thing. I'd like to offer some experience from the environment I work in at Lincoln Laboratory about system engineering. System engineering there sort of means the global job of matching a technical product if you like to a very frequently vaguely defined set of requirements. How do you synthesize the system? The role of system engineer is something that -- let's say talent as a system engineer is a very rare commodity. It almost never appears in a fresh recruit out of school. It occurs in the case of the successful practitioners as a result of a fairly extended apprenticeship in an environment where that sort of thing is done. I firmly believe that you don't go to school to learn to be a system engineer, at least not the way schools are conducted today. In a sense you'd almost want to ask the question whether system engineering is an art or a discipline, a professional discipline. I think one could make a good case for the fact that it's art. Of course there are professional and technical underpinnings of it but as it's practiced today it's very largely arty which makes me raise the question of whether or not one could do academic research in system engineering. I'm not quite sure what that means. I certainly believe in academic research in aids to system engineering; but in system engineering I'm not quite sure what that means. I think the one thought that I have or idea that I have on a role that universities can play to enhance system engineering is for the engineering schools to take a cue from perhaps the law schools and other professional schools in order to study interesting and illuminating cases. There are a lot of good, bad and indifferent communication systems if you like out there on the field. I think that it would be a very constructive thing if there were carefully prepared cases that describe the problem, the solution, its components, why certain things were done or not done, and expose that to a class for critique, for analysis, for understanding. Now preparing such a case is a horrendous task of course. But that kind of overview of communication systems problems is something that students are never exposed to until they get out into the field and work for a very long time.

BOOTON: Sol, excuse me, I'd like to comment on that if I could. I know of one case where this has been done. Victor Frost at the University of Kansas had a graduate seminar where they went through several case histories. We gave a two week discussion on the design of the payload for the tracking and data relay satellite. AT&T did one on the design of a telephone switch, for example, and these were fairly linked to discussions of what the problem was, what the process was, and what the final result was, and then the students were asked to go through their own version of this and that was very successful, but that's the only one I know of. I think that was a little bit closer to the real world and real system engineering. That's the only one I know of though.

GOLOMB: Yes, I remember about 15 years ago Si Ramo gave a lecture at Caltech with the title "System Engineering - Can It Be Taught?" and in his own lecture he concluded that the way to teach it is by the case method and then he actually gave such a course at Caltech one quarter. Ed

Rechtin's point, by the way, is that say during the Italian Renaissance and during the period of the building of the great cathedrals, no one was ever identified as an architect; it was something that from years of experience, of having been on other projects, you learned sort of what the important elements were, and then the senior craftsman who had acquired all of the relevant experience in fact became the architect; whereas today we have schools of architecture at many universities and people graduating from those institutions with relatively little on-the-job experience. Actually now most of the things that are relevant to being architects are in the building trades. So the point being that if we tried to design educational programs around this, we might be able to design system architects in engineering. In fact you might even say that the people that we call architects from architecture schools are civil engineering *architects, that that's the* trade that that grew out of.

CHETHIK: I have perhaps some disconnected thoughts on this broader subject. One of the possible interpretations of system engineering that occurred to me is perhaps a little twist on words it's what I would call systematic engineering, a disciplined and systematic way of going about satisfying a requirement with a communications system in our case. I think that there are a number of ways of codifying, if you would, a systematic approach to engineering which involves the learning of a number of tools and disciplines, some come from operations research, some come from computer technology. The other part of that, the creative aspect, I have this fantasy that creativity comes out of chaos to a large extent; it comes from chaotic thinking and thinking in ways that are unconventional. There's a kind of dichotomy, at least for me, between systematic engineering on the one hand and creativity on the other. I've never heard anybody realistically attempt to purport to teach creativity, perhaps only by example, being around some creative people, and then *it's* kind of *hit* and miss. Like I said, my thoughts on this subject are disconnected, I don't have a bottom line on this ... so I'll stop talking.

REY: Let's see, I wonder if the thought which came to mind when the statement of creativity was brought up is that I'm not sure in system engineering it's creativity that's needed. What we're doing from the very beginning is solving a problem. We're problem solvers and I think that's one of the things that good system engineers are. They know what the technology is, they have the theory, they know the methodology of designing systems, and they usually have had pretty good experience. I mean, they do go through an *apprenticeship and what they're doing* is they're solving a problem for a customer. A customer has requirements and there's a certain methodology that a good system engineer goes through to solve that problem. Now creativity is something that is probably needed, but my feeling is that's needed in the research aspects of the tools and the theory which are needed for a practicing system engineer.

REIFFEN: Certainly system engineers solve problems, but a major portion of their work is to define the problem. If we let the customer define the problem we frequently face the dilemma that the customer's definition of the problem has no solution. One of the central jobs of the proper system engineer is to illuminate the tradeoffs and say "If you want that, then you can't have that ..." or if you have a constraint on that, that imposes something else. To be able to structure the problem,

to display the hard choices, is a major function of the system engineer before there really is a consensus on what the job is to be done. Problems don't come in that crisply defined.

MOHANTY: As a matter of fact there's the book which came out early in 1970, called *System Engineering* by Porter. And there are some schools which offer M.S. degrees in system engineering or system effectiveness; and they combine in general the tools of control theory, optimization, operations research, large scale modeling, computer simulation and expert systems. I remember Simon Ramo's paper which came out in *IEEE Trans. Aerospace and Electronic Systems*, but somehow Dr. Ramo has alluded to a discipline which might be called fuzzy sets. I think system engineering is a well disciplined area and there are many good books including one by Porter on the subject. Whether it is a communications modeling or large space station design, the study involves system modeling, optimization, software development and hardware implementation. There is no room for fuzziness, but that doesn't mean there is no flexibility. So most of the time industry people when they talk about large scale or uncertain systems, probably they don't find the right word, they use system engineering. I mean my feeling is that there is a lack of exactness to define system engineering; but on the other hand, it has been a very well defined discipline in many areas, particularly at system level. There are many journals devoted completely to system theory, and I don't think those journals are any way vague or fuzzy. That's my comment, thank you.

SANDER: While they're moving the microphone ... you know, I'm sitting up here listening to what you guys are saying. There's no way I know as much about communications as any one of you, probably even your first year graduate student, but it occurs to me that it looks like I have comments coming from what I would have to term a bunch of ostriches sticking their heads in the ground. When you talk about the fact that you're engineers and, you know, you're given a problem and you can either solve it or you can tradeoff this for that; but you don't want to think creativity at all, you don't want to try to think well maybe there's something we don't have available today that somebody may be able to discover, create or invent that will allow us to solve a problem that we haven't been able to solve previously. Frankly, you know, if that's the way Americans are thinking today, there's no doubt in my mind why the Japanese are beating us in electronics.

REED: I have a couple of comments, one is on the case history method. I've never actually taught the case history method, although I actually studied law school once, and I just wanted to comment a little bit about, you know, the law school process. The case histories process that they use is actually a sequence of very, very small appellate court cases, and each case history that you study in law school is to illustrate the principles of law that you're supposed to be learning in the process of studying law. Actually I suppose that could be done in engineering, but the case histories law has usually reaped appellate court decisions, and these appellate court decisions that you read in law school are usually just a few paragraphs. The case is summarized in a few paragraphs and you attempt to, in these paragraphs, abstract the principles of law that you're about to learn. You know, it might be if a contract is broken, you're given an illustration where a contract actually was broken and how the

appellate court decided utilizing the principles of law which are commonly accepted, going back to the precedents and the common law that was used prior to that. Another difference between engineering and law or science and law is that law is, at least British and American laws, based on the system of precedence, and it usually is not based on any absolutes; whereas we in engineering, although we use inductive reasoning to invent things, we also have to use very accurate analysis to verify that what we're going to do will in fact work. So there's quite a bit of difference, and I'm not sure we can relate the two. Another comment I want to make is in this area of why communications engineering is not considered very highly by Congress. For some reason or other we as communications engineers have limited our scope in communication engineering. We think more in terms of the simple channel, you know, the so-called Shannon channel, and we don't get outside of that channel at all, and to illustrate that you can go beyond the simple channel, in fact we've gone beyond the simple channel; of course we're interfacing with the human communication problem, which I know if you talk to any psychologist there's a great deal of interest in communications. In fact if you talk to any congressman you'll find he's very interested in human communications. So I think that we have to somehow get across the principle that even human communications, which everybody knows is important, can be enhanced by communication engineering, and that ultimately the purpose of communication engineering is to enhance the capability of human beings to communicate with one another so that they can communicate reliably so they don't make mistakes in the process of communication, and to lead to better understanding. I think that if we to some extent broaden our scope, perhaps we would be better able to sell the idea that communications is a very important subject. And I agree with you, Bill, that we have to do more than what we're doing with respect to the government and industry to present a better picture of what communication engineering is.

GOLOMB: Thank you .... Gaylord ....

HUTH: Let's see, I guess I wanted to address a couple of things. Sometimes I think that communications system engineering and creativity are separate issues. I say that for the following reason, it has been my experience that you're able to be creative when you are working on a fairly vaguely described topic and have the freedom to develop a general solution. What happens in system engineering as it's practiced in industry, at least from my experience, is that the research stage is past and while the project may be vaguely defined, because of the contractual structure, they're going to building something and it's going to be built at the minimum cost. So what happens is that as a discipline, I think Raul was describing the procedure that is followed. There isn't any room left or very little room left for anything that's really innovative at that stage. It is now like we're going to freeze technology and we're going to solve these problems. Therefore, it's a problem solving situation rather than doing research. The research was done under IR&D, or whatever, previous to when you got to the contract, or there was other research looking longer term. So there are actually two different things. So I think that Raul was probably talking more about the contractual problem of how you actually implement something rather than necessarily the previous research that enabled you to do it. I do think, though, as

I've seen from the students coming out of school, there are two groups. There's a group that comes out of school taught how to do research, or at least that's what they were doing especially Ph.D. students, because that's what they were supposed to do. They're supposed to do research and get a dissertation approved. When they go to work in industry they expect to do research. There seems to be a small number of jobs in research and a large number of jobs doing implementation of systems. But I think that in some cases it would be nice somewhere along the line to have an idea of what you were going to get into when you work in industry, where do you fit in and how do systems get created in industry. I think that we can do the apprenticeship, but I think it would be nice to have something like senior problems where you at least get an idea of what industry expects of you. Maybe you go through some case studies just so you get an idea. The course would give you a flavor of what you're going to have to do when you're out on the job. I can see a semester course where you get familiar with what your environment is going to be and how you're expected to operate in it.

GOLOMB: Yes, I guess I have a couple of quick comments. One is a principle that I formulated a number of years ago, I guess we'll call it "Golomb's Law". This is what the difference is between pure research and applied research. And it turns out that pure research is what *you* want to do, and applied research is what someone else wants you to do. [laughter] The other observation that I have is that there's been a lot of discussion about what is creativity? What can you embody in the computer program that assists or even takes over the task of the system engineer? A dozen or so years ago we had an assistant professor in the Math Dept. at USC who was much interested in designing computer chess playing programs. He had been National Junior Chess Champion and sort of retired from active chess with a rating of Senior U.S. Master, and had he been in international competition he probably would have had a Grand Master rating. He was also a personal friend of Bobby Fisher. I remember having discussions with him about this view of mine at that time when chess playing programs were much more primitive than they are today, was "Yes, well I could kind of see how you could program a computer to play at the level of a C player, maybe even a B player." At that time I think I had an expert rating myself, so it was sort of felt "Well obviously what I do couldn't be taught to a computer, but maybe about two grade levels below that." So Charles Kalme, who was the chess programmer, said yes he had thought about this and he asked Bobby Fisher on one occasion what Fisher thought, and Fisher said, "Yes, there's a certain level of play that's pretty mechanical and you could program a computer to that level." He asked Fisher what Fisher thought that level was, and Fisher said, "A Grand Master." [laughter] I think there's a great deal of truth in that. The fact is today the best chess playing programs are playing at or near the Grand Master level. I think there's always a certain amount of skepticism about what computer-aided decision making can do when you're in the early phases of approaching a problem. I'm sure the first attempts at computer-aided diagnosis were coming up with results, say as, "This man is pregnant," you know, because it didn't quite have all of the necessary constraints on what are reasonable solutions. But given enough time and enough sophisticated programming effort, I think we do

end up at a situation where we can get computer decision-making which is better than that of many of the people who are actually out there in the field who are supposed to have been doing that sort of thing. Lloyd ....

WELCH: I have a problem with that in that you have a pretty well defined criteria of how well a chess program is doing -- you play it against another chess program, then there's a winner or a draw. In system engineering, you know, when you get done you have a system, but you don't nearly know whether it's the best system, optimal ... in other words, there's the lack of criteria here for how good a system engineer has performed his job.

SCHOLTZ: I think I've been trying to wrestle with a problem for about a year that we haven't really discussed yet and that's this question of university/industrial interaction and what's the best way to foster it. I thought this workshop was going to be it, and perhaps to some extent it has, but now that I want to raise the question I think most of the industrial people are already gone. I see mostly academics here, a few TRW people, Lincoln Lab. I'm interested in TRW's experience and maybe that of the few other people here with regard to not only what was successful but what was unsuccessful in university/industry relations. What should we really be aiming for in trying to develop this university/industrial cooperation in research and in education? I'll be darned if I know exactly what I would put in a systems engineering discipline degree at a university at this point. Maybe you people also have ideas on that, besides case studies. Somehow case studies remind me of perhaps trying to prove something by example; I don't know that the analogy is quite correct. It's interesting and you learn from it but I don't know whether that's enough to make a discipline in any real way.

REY: One of the things that I think we've got to be careful of, and this has been the age-old problem, of having universities set up curriculum to meet industry's needs. I think there's a real problem there; probably being in industry I should want to force you in the other direction. But I think it's Gresham's Law that industry is much more interested in short term results, and they have immediate problems and they're looking at their immediate staffing needs for hiring system engineers, and they would like a certain kind of system engineer because next year they see these kinds of projects coming. I think that the university has to be somewhat independent in looking at technology and research and training engineers in math, physics and theory, communications theory, which allows them to solve problems. To teach the process of system engineering you could have a course that says something about the methodology, but I just don't see a degree in that. At TRW where we really tell people what they're going to be as system engineer, is in the interview process. Right away we tell them, "You're not going to do research; if you want to do research, you better go somewhere else, because you're going to be in the high bay and you're going to reduce data ...." And we've just decided we tell them that right up front, otherwise they're going to become disillusioned in just a year. The kind of interaction that I think has worked out real well with universities is our Fellowship Programs where we look at the kind of person we want, the type person, the training they've had in the past, give them a fellowship, and send them to a university like USC, and we also have a program like that

with Kansas. I think that's another real good example of industry/university interaction. The BOSS effort, doing some sort of basic, I guess I wouldn't call it basic research, but conceptual research on the next tool that we needed was a good example. I guess I'm kind of concerned since I've seen other organizations trying to create system engineers. They'll go out and hand them MILSPECS and textbooks on system engineering and they don't get system engineers because they missed one important factor of system engineering, and that's the understanding of math and the physics. Those are the people that are hardest to get, the ones who understand the mathematics and the physics, and they usually do much better in system engineering than people trained in the methodology.

GAGLIARDI: I just wanted to say that I agree wholeheartedly with those comments. Systems engineering as we see it is a combination of knowledge and experience, and you can't teach experience. All we can do at the university is teach knowledge -- we can't even do that, all we can do is state facts. And that's our role, to state facts in the classroom and make sure that we update as often as we can.

BOOTON: Sol, I think I've got a comment ....

REED: I think we're teaching a way of thinking more than the facts. There is some analogy between a law school and science. Actually what they do in a law school is teach you a way of thinking, and that's really what we're trying to do here, is teach people a way to think, a way to create. Actually there's also an analogy to law in a sense; to invent things you have to do what lawyers do, you have to learn what is called the inductive reasoning process. You have to create the case so to speak, and that reasoning process I think has to be developed in engineering and in science just the same way as it has to be developed in law. In law there are more human constraints. Law has a structure, it is more like the chess game where there are rules of the game; whereas here, there are rules of the physical laws of the world. I claim that there is this analogy that it's a way of thinking that we all have to learn.

GOLOMB: I might mention the case method is used in many disciplines other than law. It's used in business schools, it's used in schools of public administration, and in each discipline for the case method to be effective it has to mirror what it is that you're ultimately trying to do in that profession. In law you are ultimately trying to win a case in court which is unique to law and is not common to others. Yes, but I'm saying that in terms of what the bottom line is. The bottom line in engineering is very different, it's not to win a case in court, it's to build an effective engineering system and so on. I think that Bob Scholtz though has really raised one of the most important things to focus on which is what are the areas of meaningful cooperation that can occur, in particular between universities and industry, without getting too bogged down by issues of company proprietary things. I think peripherally this conference has touched on a lot of it, but you might almost have to form a special kind of workshop or something just to address the details of that issue.

BOOTON: Yes, Bob asked two questions that I'd like to make my comments on. One of them was this question about what can you teach about system engineering. I think you can teach a lot, not the experience, but there are things you can teach about how to engineer a communication system. In the environment of

communication systems that I am talking about, I'm very skeptical about general system engineering. Some universities have tried to give general systems degrees; we wouldn't hire them (people with those degrees). The different aspects of system engineering in various areas sound the same and seem to have a lot in common, but a communications system engineer first of all has to know communications. These general engineering or general thinking things can be taught but the graduates of most of the general systems degrees so far haven't learned enough about any field, the logic, operational research, and so on. I think you have to first and foremost teach about communications.

The other question was about what's going well and what's doing poorly as far as industry/university cooperation is concerned. The things that I've seen to do better are in those areas in which first of all there is at least one strong faculty member at the university and one strong engineer at the company that are working jointly on something and have a strong interest. They have to be working on something that the company cares about. A lot of company-funded research at universities is sort of money tossed over the fence as far as someone to work on a Master's degree or a doctoral dissertation, and no one cares about the results. Only when the company really cares about the results and hopes that something useful comes out of it seems to work well. I think that was the case in this BOSS effort, though we've had two or three others; but in a lot of them there's not enough serious work in trying to make the process work in the strong personal tie.

GOLOMB: I know from the position I've had the last year or so as Vice Provost for Research that the thing that contractually universities are terrified of is the notion that they will enter into a relationship with industry where they actually promise to deliver something that works. [laughter] I also have a comment about education. I'm not convinced that teaching exists at all. I'm convinced that learning exists and that the role of the institution is to try to create an environment that fosters learning. So really the question of whether you can teach system engineering I think is part of the broader question of can you teach anything? The student has to learn and it's not clear to me that you can force that process without the student's participation. Let's see, did we have other comments?

REED: I just wanted to make a comment about the relationship between industry and university, and also I guess this is related to the government relationship to university and industry as well. When I was a young man just coming out of the service, going back to Caltech, I had the ambitions at that point, because I'd been in the service and I'd worked on electronics, the new thing was electronics. So I visited with Royal Sorenson, Prof. Sorenson at that time was the head of the Electrical Engineering Department. At that time there was very little support from the government for electrical engineering, in fact I don't think there was any, and I spent quite a bit of time talking to him about this as I remember. Apparently most of his support had been obtained from his work with the Edison Company, prior to the war during the years where they built Hoover Dam or Boulder Dam or whatever name you want to give it now in the transmission lines. He actually was a consultant to Edison Company, and much of what he did as a consultant was apparently assign development projects for many of his students right from his own consulting and this was, I would say, a very intimate connection with indus-

try and university. He created projects for his students through this university connection. I don't know whether that's practical today or not, but I still think that a system like that could be used today with industry by professors which would help supplant this problem we now have with the government, apparently with the Congress that is not close enough to the real needs of industry and government, particularly DoD, in understanding how desperate the university is for funding at certain times.

SANDER: I think the government and Congress understand the need for funding at the universities; I think that was one of the reasons why they changed the URI line from the services to the DoD. But the point I was making was I can't see an acceptance of the fact that communications research for Army-specific problems is essential to the survival of the Army in the battlefield. I went up in December, I was in the office of the Assistant Secretary of the Army for R&D, or the Director of Army Research Office, not ARO but the Office of the Director of Army Research, to write a speech for him. He had to give a speech to the Association Defense Preparedness Agency or whatever. On two iterations of that speech that I was writing for him, I put in viewgraphs on communications. I got it back, after the first iteration, and he had put big X's on those viewgraphs for communications. So I asked the guy that was working for him why, and he said "Well, because he doesn't understand communications." So I wrote a much simpler description, something that my son probably could have understood about the problems of communications in the Army and why we needed special programs in that area, put it back in the second iteration, and wound that up and went home. About a month later I got a copy of the final draft; there was absolutely nothing in there on communications, right. So, you know, I'm convinced that they simply do not understand there is a problem, and I'm talking about Army. In regard to Bob's question about how industry and a university interact and work, there have been a number of examples and I think that USC-CSI is one of them, in which government, industry and university all had a role in doing something worthwhile. Some years ago, I guess, two different locations in the country, the Army Research Office under the University Research Instrumentation proposal gave equipment money -- and Bob, I think you got some equipment money from us, didn't you -- okay, so Bob was one, the other one I know for sure is Misha Schwartz at Columbia University. Now this was seed money and it turned out to be highly leveraged. Because of the result of the research funding that we have been supporting at USC and the equipment, you know, the instrumentation money at USC, they were able to put together a little bigger package with some help from industry and form the CSI, and then they evolved. I don't know whether you have an industrial affiliates program now or not, but this is getting bigger, it's getting wider exposure, they are there to help industry solve problems and industry is becoming more aware of that and participating more in the programs at the CSI. The same thing is happening at Columbia University with Misha Schwartz's program, and Schwartz got one of the NSF contracts for a Center of Excellence in Communication. One of the ways you do this is you have a review every year, just like Bob does, and industry people come in and listen to that review and you have an opportunity for exchange of ideas, feedback and directions being gotten. There you have a very good example of the

university, industry and government cooperation. The government looks at funding something like that as leveraging; you put in seed money to get the leverage. Another example has been the development of CAD software. SPICE was developed at Berkeley; SUPREME is an example of a little program developed at Stanford. Because the majority of the support for the development of those programs came from government, they were nonproprietary. Industry all over the country, all over the world, picked those programs up and started utilizing them. Then the universities starting getting feedback from industry as to where the programs had shortcomings, where they didn't perform and do what they wanted them to do specifically. The universities then started doing research in those areas. You found the universities started having reviews -- I know Stanford has a one week seminar every summer to go over, their progress and status of their programs in CAD for microelectronics. I don't believe Berkeley has that, but Carnegie-Mellon has that, and Carnegie-Mellon in the CAD of IC's also has an industrial affiliates program. And to join you have to pay money. Now the government has even started doing that themselves. I don't know how we would do that directly from an Army laboratory and I haven't heard of a citation where anybody has done that directly from a government laboratory. But what the Army laboratories have done, and I have heard of one or two examples of that is, they MIPR money to NSF, and a MIPR is a way of transferring funds between government agencies, I don't know what it stands for but, MIPR money to NSF and NSF apparently can write a grant or a check to one of these industrial affiliates for the laboratory to join as well. I know the government, DoD, paid $1,000,000 to join SRC this past year. I want to say that in the last few years, we've seen a great growth in the industry/government/university cooperation, in mechanisms and expansion of the number in types of mechanisms for doing it.

SASTRY: I have background both in teaching as well as industrial research, so I thought having been here I should say something to contribute to the discussion. The system engineering, in our profession system integration perhaps, also involves exploration of alternatives basically, and I suppose we all agree on that. But in the main industrial environment as rightly was pointed out by several (people), sometimes you are conditioned by state-of-the-art, maybe for the next two or three years, in exploring other alternatives. So these alternatives can of course be pursued by the university. But in the industry, though people cannot pursue it, it doesn't mean they are not aware of it. They simply don't have the time or facilities to do it, or the forum to do it, so they'll have to interact with the university to do that. So that awareness somehow has to be folded into this interaction that I think is the crux of the problem. How to do that? Well when we say interaction with the university, whom do you interact with -- are you talking about interacting with faculty or are you talking about students? I happen to believe that when you are looking at these alternatives or exploring alternatives and comparing them, the most important ingredient is the availability of individuals with right backgrounds, and how wide are their backgrounds. So basically what I'm getting at is you need an individual who has a good background. I don't see any alternative to that like students not having in-depth courses in electromagnetics, in

mathematics, in communications theory, modulation theory, noise, in computer systems, and all those things which are building blocks. I see a dangerous tendency where backgrounds of most students are being narrowed down in terms of their expertise and especially by the time they come to a senior year it's becoming fashionable to work on a project which is, in my view, a very narrow thing. What we need for the students really is help motivate them to stick to the fundamentals. They keep looking at equations, solutions, matched filter theory and all of that but do not know where to use them. Before they get turned off by those things we should continuously keep them motivated. I think we have to do that, not just in the final year but maybe in the last three of the four year course. They should be taken on field trips perhaps. Take them to your local broadcast transmitter, tell them right there how it works. I used to do that for a 15-day period, taking the students in different types of places and the reaction you get is really very interesting. Otherwise, they'll do amplitude modulation and all that, all the equations, and they don't know if in order to transmit from here to a hundred mile range you need W, KW, MW. But once they see that this sort of thing is going to be used in the future, they will concentrate more on learning the fundamentals. I don't see any alternative for the learning of the fundamentals. You can call it engineering. So basically what I'm trying to say is that finally today's students are tomorrow's systems engineers, not today's systems engineers. Who is then going to contribute from the universities? Obviously faculty members with good and wide backgrounds can carry out (this task) by interacting with the industry and passing on their personal experiences as part of their lectures perhaps, quoting situations, so-called case studies for example at a particular point. Maybe, say, you are teaching frequency modulation or something and you found by practical experience in interacting with some industry that some problem was solved. That is the spot to tell them a few minutes how this is accomplished. I don't how well the courses in case studies are practical projects or something like that are going to help. I think they are a waste of time. In those four years an individual can learn a number of things. After they enter the professional life, they may not have time again to learn the fundamentals with all the so-called continuing education and all that. So let us concentrate on fundamentals for the students when they are at the university and let faculty take more initiative on so-called system engineering issue, passed on to them blended in the class instruction.

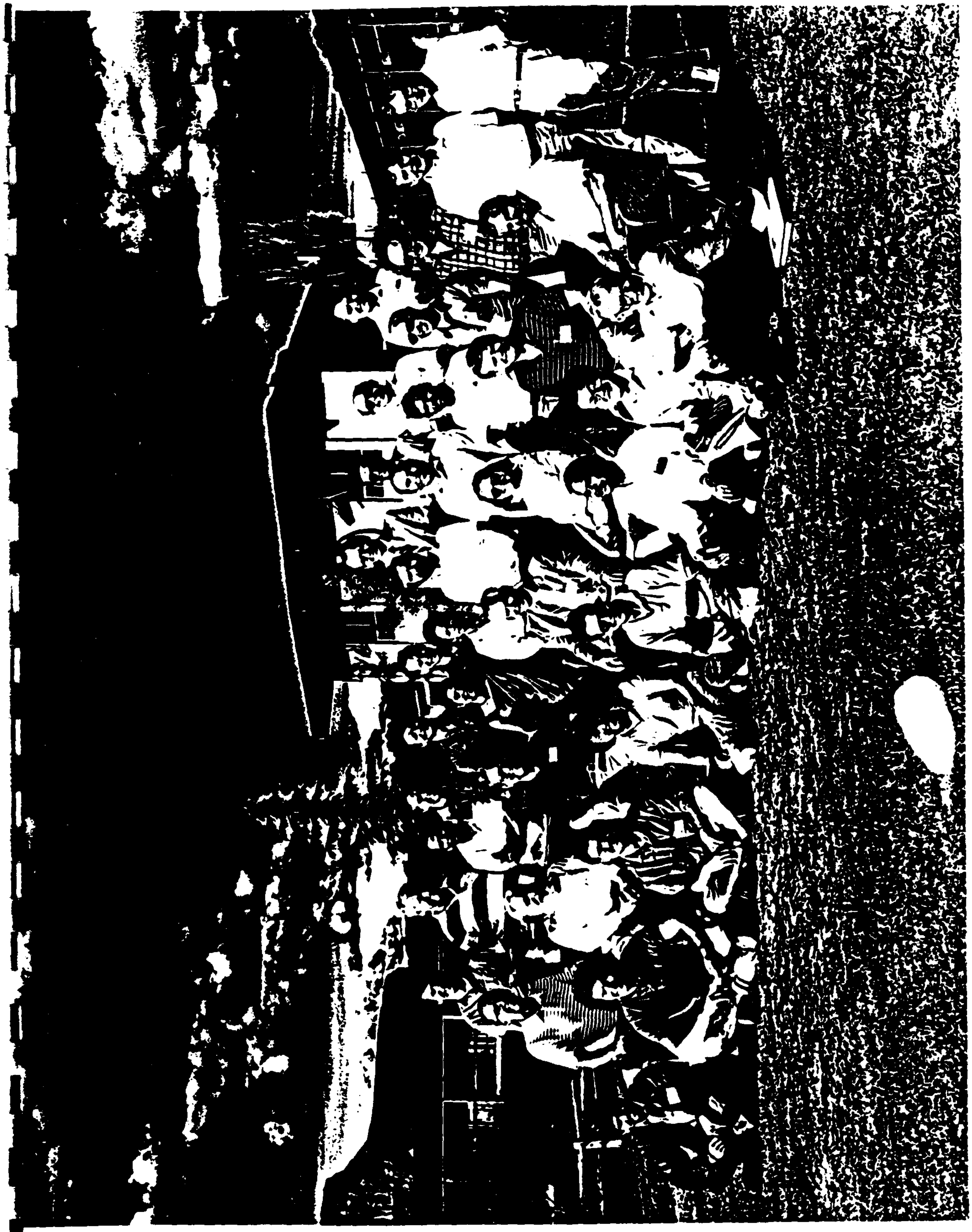
GOLOMB: Thank you. I think Gaylord has a comment ....

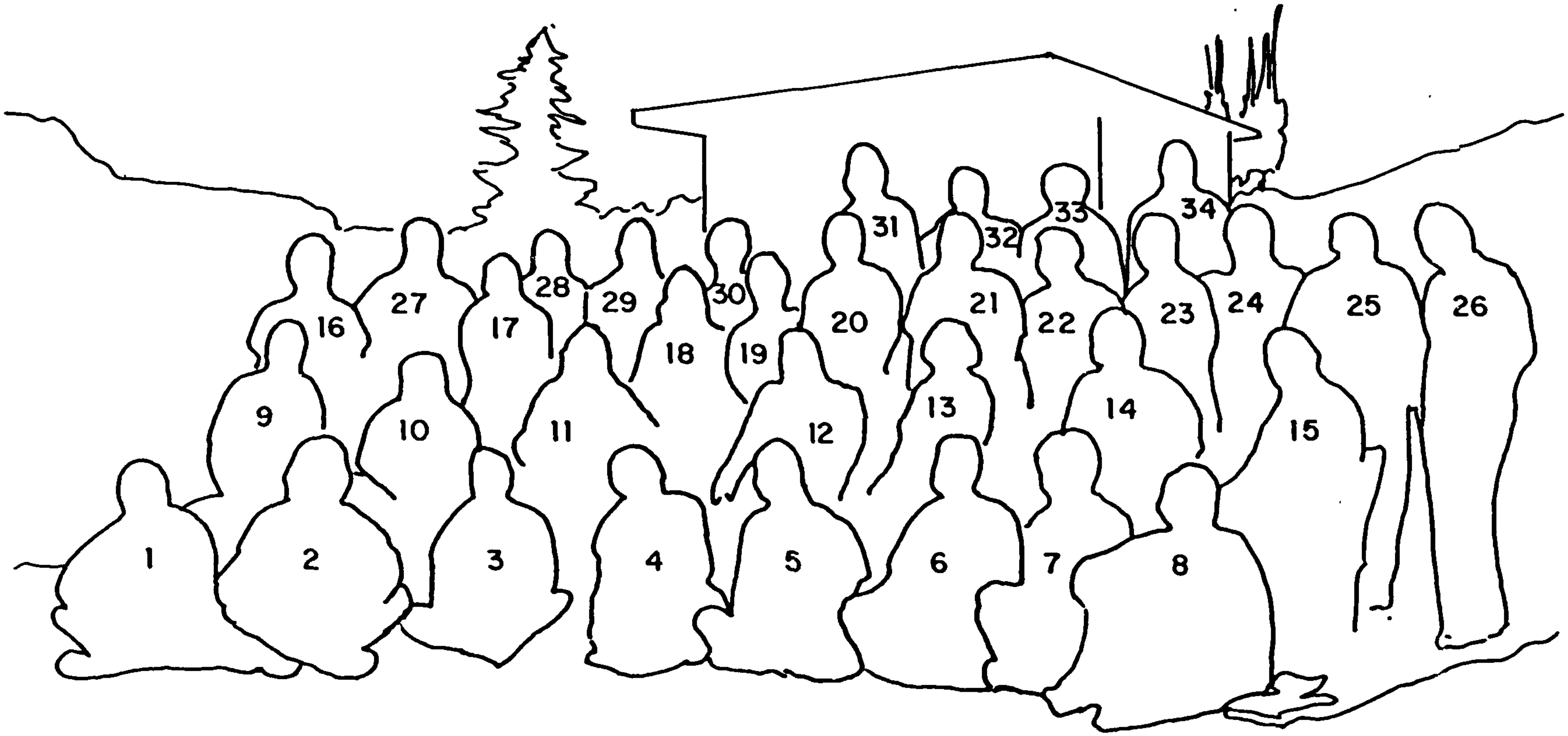
HUTH: Let's see, I am pretty much in favor of describing to the future employee, the relationships between him and TRW. Raul already does this to a large extent. Raul mentioned another relationship between the university and industry. While somebody is working in industry, they can also be going to school and taking courses. This worked well for me. I worked all through graduate school and I was able to apply what I learned in the university very quickly to my work, and alternately my work provided me with a framework to what I was supposed to be learning. It was a very, very good environment. I think the combination is very important and it is a good relationship between the university and industry. I think that sometimes when students go straight through the university with no work

experience, they don't get as valuable experience as if they had traded off a little bit between work and school. USC already does this to a large extent so I don't know whether there's something that I can add to your environment, but I think it is a very important one. This combination of working and going to school in some sense satisfies my thing about senior problems because you're already in industry and you're seeing these things happen. You're learning system engineering and you're finding out the courses which will be most valuable to you. In my first job at Hughes, I was a circuit designer and that was all fine except I didn't know how to make circuits stable because I hadn't taken any control theory. So control theory was one of the first courses I took at SC to find out how to make these circuits stable. I learned a lot that way.

GOLOMB: Are there any other comments? Maybe I'll turn it back to Bob ....

SCHOLTZ: I think at this point I can only thank everyone who stuck this out to the bitter end. We were really pleased to have you here, I hope you enjoyed yourselves, and I'd like to assure you that I hope that you'll all be at the next workshop that CSI sponsors, and we're very happy to have you. Thank you very much and the meeting is adjourned. [APPLAUSE]





27. Gaylord Huth AXIOMATIX; 28. Charles Wheatley MA/COM; 29. Bob Gagliardi USC; 30. Fletcher Wicker Aerospace; 31. John Choma USC; 32. Ali Zolfaghari Pacbell; 33. Ray Pickholtz George Wash. U.; 34. Charles Bosco ARO

16. Andreas Polydoros USC; 17. Nirod Mohanty Aerospace Corpn.; 18. Milly Montenegro USC; 19. Cathy Cassells USC; 20. Mike Pursley U. Urbana; 21. Carl Ryan Motorola; 22. A.R.K. Sastry Rockwell; 23. Randy Reitmeyer ARO; 24. Frank Chethik Ford Aerospace; 25. Raul Rey TRW; 26. Jim Rheingans Hughes

9. Robert Scholtz USC; 10. Solomon Golomb USC; 11. Lloyd Welch USC; 12. Irving Reed USC; 13. Carolyn Jack GTE; 14. William Sander ARO; 15. Jim Spilker Stanford Telecomm.

1. P. Vijay Kumar USC; 2. Chuck Weber USC; 3. Gregory Yovanof USC; 4. Jim Kurose Amherst; 5. Nikos Pronios USC; 6. K.C. Gupta U. Colorado; 7. Bing Sheu USC; 8. Dick Booton TRW

WORKSHOP - ATTENDEES

May 26-29, 1987

Sedona, Arizona

Dr. Frank Amoroso (714) 441-9661
Hughes Aircraft Company
Bldg. 675, MS R221
P.O. Box 3310
Fullerton, CA 92634

Dr. Philip Balaban
Computer-Aided Design Department
Room 2E516
AT&T Bell Labs.
Holmdel, NJ 07733

Dr. Barry Bertiger (602) 732-2272
Motorola Inc. or 3084
Strategic Electronics Div. G 2229
2501 S. Price Road
Chandler, AZ 85248-2899

Dr. Richard Booton, Jr. (213) 812-2103
TRW Electronics & Defense
ESG
Mail Code E2/3080
One Space Park
Redondo Beach, CA 90278

Commander
U.S. Army LABCOM
Attn: SLCET-IC (Charles Bosco)
Fort Monmouth, NJ 07703

Mr. Don Calhoun (714) 441-9545
Hughes Aircraft Company
Ground Systems Group
P.O. Box 3310
Bldg. 688
Fullerton, CA 92634

Mr. Frank Chethik
Staff Scientist (415) 852-4690
Special Programs Operations
Ford Aerospace &
Communications Corp.
MS X30
3939 Fabian Way
Palo Alto, CA 94303

Ms. Cathy Cassells (213) 743-8306
USC PHE 508

Dr. John Choma (213) 743-8776
USC PHE 604

Dr. James DuPree (213) 297-8651
MS R10/2747
TRW Electronics Systems Group
One Space Park
Redondo Beach, CA 90278

Dr. Robert Gagliardi (213) 743-5540
USC PHE 418

Dr. Solomon Golomb (213) 743-2990
USC PHE 506

Dr. K.C. Gupta (303) 492-7498
Professor of Electrical Engineering
University of Colorado
Campus Box 425
Boulder, CO 80309-0425

Dr. Dale A. Harris (415) 823-3060
Executive Director
Strategic Technology
Pacific Bell
22600 Camino Ramon
Room 3S453
San Ramon, CA 94583

Dr. Gaylord Huth
AXIOMATIX (213) 641-8600
9841 Airport Boulevard, Suite 912
Los Angeles, CA 90045

Carolyn Jack (617) 466-2915
Member of the Technical Staff
GTE Labs Room 1-224

40 Sylvan Road
Waltham, MA 02254

Dr. Michel Jeruchim (215) 644-7307 (H)
Senior Communications Engineer
General Electric Company
Space Systems Division
Building 9, Room 2123
P.O. Box 8048 (215) 531-7197
Philadelphia, Pennsylvania 19101

Dr. J. Jay Jones (415) 852-4699
Technical Consultant
Ford Aerospace & Communications
3939 Fabian Way MS-X10
Palo Alto, CA 94303

Dr. P. Vijay Kumar (213) 743-5387
USC PHE 416

Dr. Jim Kurose (413) 545-1585
Assistant Professor
University of Massachusetts, Amherst
Department of Computer
and Information Science
Lederle Graduate Research Center
Amherst, MA 01003

Dr. Victor Li (213) 743-5543
USC PHE 408

Gregory C. Marr (303) 570-8888
1450 Academy Park Loop
Suite 200
Colorado Springs, CO 80910

Dr. Nirode C. Mohanty (213) 336-4220
Manager, Communication Modeling
The Aerospace Corporation, M4-937
P.O. Box 92957
Los Angeles, CA 90009

Milly Montenegro (213) 743-5537
USC PHE 404

Professor H.T. Mouftah (613) 726-3380
Department of Electrical Engineering
Queens University
Kingston, Ontario
CANADA K7L 3N6

Prof. Ray Pickholtz (703) 280-5628 (H)
Electrical Engineering
and Computer Science
George Washington University
9613 Gleenbrook Road (202) 994-6538
Fairfax, VA 22031

Dr. Andreas Polydoros (213) 743-7257
USC PHE 414

Dr. Ken Porter (602) 949-3123
Motorola, Inc.
P.O. Box 1417
Mail drop H 3122
Scottsdale, AZ 85252

Nikos Pronios (213) 743-5531
USC PHE 424

Dr. Mike B. Pursley (217) 333-2966
3-121 Coordinated Science Laboratory
University of Illinois
1101 West Springfield Ave.
Urbana, IL 61801

Dr. Irving Reed (213) 743-2991
USC PHE 510

Barney Reiffen (617) 863-5500 x7600
MIT Lincoln Laboratories
P.O. Box 73
Lexington, MA 02173

Commander (201) 544-3465
Division Director, Microelectronics
ETDL, LABCOM, SLCET-I
Attn: Randolph A. Reitmeyer
U.S. Army LABCOM
Fort Monmouth, NJ 07703

Raul Rey (213) 297-8632
TRW Electronic Systems Group
R10/2731
One Space Park
Redondo Beach, CA 90278

James M. Rheingans (714) 441-9623
Senior Scientist
Systems Engineering Laboratory
Hughes Aircraft Company
P.O. Box 3310, Bldg. 675/C212
Fullerton, CA 92634

Dr. Carl R. Ryan (602) 732-3074
Government Electronics Division
Motorola, Inc.
2501 S. Price Road
Chandler, AZ 85248-2899

Dr. William A. Sander
SLCRO-EL (919) 549-0641
Electronics Division
Department of the Army
Army Research Office
P.O. Box 12211
Research Triangle Park,
NC 27709-2211

Dr. A.R.K. Sastry (805) 373-4409
Science Center A35
Rockwell International Corporation
1049 Camino Dos Rios
P.O. Box 1085
Thousand Oaks, CA 91360

Dr. Robert Scholtz (213) 743-5546
USC PHE 534

Prof. Sam Shanmugan (913) 864-4832
Telecommunications and Information
Sciences Laboratory
2291 Irving Hill Road,
University of Kansas
Lawrence, Kansas 66045-69291

Dr. Bing Sheu (213) 743-0596
USC PHE 604

Dr. James J. Spilker, Jr. (408) 980-5600
President
Stanford Telecommunications Inc
2421 Mission College Blvd.
Santa Clara, CA 95054

Dr. William Steier (213) 743-0149
SSC 502

Mr. Dan Sullivan (213) 812-1458
E2-10063
TRW Electronic Systems Group
One Space Park
Redondo Beach, CA 90278

Dr. George Turin (415) 643-8435
University of California at Berkeley
Department of Electrical Engineering
and Computer Sciences
518 Cory Hall
Berkeley, CA 94720

Dr. Chuck Weber (213) 743-2407
USC PHE 402

Dr. Lloyd Welch (213) 743-2699
USC PHE 502

Dr. Charles Wheatley (619) 481-2582
Assistant Vice President
M/A-COM Govt. Systems
13983 Recuerdo Dr.
Del Mar, CA 92014

Dr. Fletcher D. Wicker (213) 336-7024
Director
The Aerospace Corporation
P.O. Box 92957, M4/937
Los Angeles, CA 90009

Gregory Yovanof (213) 743-8208
USC PHE 518

Mr. Ali Zolfaghari (415) 823-3060
Technology Consultant
Pacific Bell
22600 Camino Ramon
Room 3S450
San Ramon, CA 94583

ERRATA SHEET

Page 8 ..... not to be found in the manuscript, is not missing. It just never was numbered.

Page 103 .... appears twice. It was corrected by adding 103-a to the second occurrence.

Page 194 .... also appears twice. Also corrected with 194-a.

MED

8